



Escola Politècnica Superior
d'Enginyeria de Manresa

PROJECTE FI DE CARRERA

**SISTEMA DE CONTROL i
ESTABILITZACIÓ AUTOMÀTICA
PER A
VEHICLE AERI NO TRIPULAT**

Autor: Jordi Segura Clares

Director de projecte: Ricard Sanahuja i Moliner

Enginyeria Tècnica de Telecomunicacions

Especialitat en Sistemes Electrònics

Manresa, 11 de Gener de 2011

AGRAÏMENTS

Voldria estendre aquests agraïments, no només a qui m'ha ajudat en aquest projecte fi de carrera, sinó a tots els que ho han fet al llarg de tota la titulació.

D'aquesta forma, en primer lloc gràcies a tot l'equip de professorat de l'escola de Manresa que, al llarg de tot aquest temps han transmès o almenys han intentat transmetre el seu coneixement.

Gràcies especials als meus mestres i companys del Club de vol d'Olius els qui m'ajuden i m'empenyen tant desinteressadament en aquest apassionat tema.

I encara més, a la família i amics que han estat a l'alçada per a suportar-me i millor encara, animar-me en tota aquesta empresa.

Gràcies.

ÍNDEX

1.	INTRODUCCIÓ.....	5
2.	OBJECTIUS	7
3.	VANT (Vehicle Aeri no Tripulat)	9
4.	AERODINÀMICA d'ESTABILITZACIÓ i CONTROL.....	12
4.1.	ELEMENTS MECÀNICS i ELECTRÒNICS DE CONTROL	16
5.	DESCRIPCIÓ GLOBAL del SISTEMA	18
6.	MODES DE FUNCIONAMENT	20
6.1.	CONTROL MANUAL.....	21
6.2.	ESTABILITZACIÓ AUTÒNOMA.....	22
7.	DESCRIPCIÓ dels COMPONENTS del SISTEMA	24
7.1.	ELEMENTS de l' ESTACIÓ TERRA	24
7.1.1.	Ordinador personal	24
7.1.2.	Palanca de comandament (<i>Joystick</i>)	25
7.1.3.	Programari de control de l'estació terra	26
7.1.4.	Maquinari de l'estació terra.....	28
7.2.	TRANSCEPTORS del SENYAL de RÀDIO	31
7.3.	ELEMENTS del VEHICLE.....	34
7.3.1.	Actuadors (Servos)	35
7.3.2.	CPLD – FPGA.....	38
7.3.3.	Microcontrolador	39
7.3.4.	Sensor de Posició (Acceleròmetre).....	46
7.3.5.	Esquema i Muntatge del maquinari al vehicle.....	49
7.3.6.	Programari per als diferents components del Vehicle.....	51
8.	DESENVOLUPAMENT i FUNCIONAMENT.....	53
8.1.	PROTOCOL DE COMUNICACIONS	53
8.2.	PROTOCOL RS-232 (Recommended Standard 232).....	56
8.3.	RECEPCIÓ DE TRAMES AL MICROCONTROLADOR	58
8.3.1.	Obtenció de la trama de comunicació.....	59
8.3.2.	Validació de la trama amb el checksum.....	59
8.4.	PROTOCOL MICROCONTROLADOR – FPGA	60
8.4.1.	Enviament dels valors a la FPGA	61

8.5.	GENERACIÓ del SENYAL PWM	62
8.6.	Disseny del circuit en VHDL.....	64
8.6.1.	Generador del senyal de clk de 102,4 KHz.....	66
8.6.2.	Màquina – Registre <i>Adr</i> i DeMultiplexor.....	67
8.6.3.	Procés modulador del senyal PWM	67
8.7.	PROCÉS d'ESTABILITZACIÓ AUTOMÀTICA.....	69
8.7.1.	Funció de correcció.....	70
8.7.2.	Inicialització de l'acceleròmetre	72
8.7.3.	Lectura de l'acceleròmetre.....	73
8.7.4.	Càlcul d'estabilització i modificació de la trama	74
9.	CONCLUSIONS i LÍNIES FUTURES DE RECERCA	76
10.	BIBLIOGRAFIA.....	78
11.	ANNEXOS	80

1. INTRODUCCIÓ

La meva afició per l'aviació, arriba de la mà de l'aeromodelisme, que comença fa ja uns quants anys. De forma paral·lela, sent encara ben jovenet, prenc també contacte amb la informàtica quan aquesta encara estava limitada, de forma particular, a les màquines com el VIC-20 de la marca *Commodore*. Parlem de l'any 86...

Encara abans però, ja havia passat l'electrònica per les meves mans. A casa havia ajudat al pare muntant algun "kit" de la marca *Sales-kit*, i posteriorment, ja vaig ser jo solet amb algun company de col·legi qui n'havia adquirit algun per investigar. Encara recordo com un professor de sisè de bàsica es va interessar per algun d'aquests muntatges fet per nosaltres, i no el vaig veure mai més...

Això ja deixa entreveure que la tecnologia havia d'acabar formant part de la meva vida, d'una forma o altra.

No obstant, sempre he volgut trobar aplicacions interessants o més serioses en tot allò que faig. El fet de volar avions en un camp de vol, aviat es fa avorrit, si no hi ha alguna cosa més. Sempre pensava "*però aquests aparells no poden fer quelcom de més?*"

És per tota aquesta barreja de circumstàncies que, ja més recentment, llegint sobre el tema, conec el món dels UAV (*Unmanned Aerial Vehicle*), els vehicles aeris no tripulats. De fet, la mateixa idea que jo havia tingut, evidentment havia passat ja pel cap de molta altra gent i molt abans. Justament per això ja fa temps que existeixen aquest tipus de màquines, de moment, pràcticament limitades al món militar, obrint-se pas poc a poc cap al món civil.

Comença així una etapa en la que augmenta el meu interès per aquest tipus de models a petita escala tot i ser conscient de les limitacions. Però, per què no buscar aplicacions, adaptant vehicles aeris no tripulats per a missions civils reals, abaratint costos dels vehicles aeris convencionals i reduint el risc per als pilots?

Doncs gràcies als coneixements adquirits durant els anys, evidentment complementats a través de la meva titulació, pretenc iniciar o almenys investigar sobre el desenvolupament d'una petita part de tot el que hauria de tenir una màquina com la descrita.

Durant els meus estudis, he volgut aprofundir en coneixements relacionats amb l'electrònica digital, els microcontroladors i el maquinari programable com els CPLDs i FPGAs, precisament per les possibilitats i facilitats que poden, per exemple, ser aplicades en camps com els de l'aeronàutica, i més en concret, els Vehicles Aeris no Tripulats.

És per tot això que, quan arriba el moment de realitzar el projecte final de carrera, vull aprofitar l'ocasió per aprofundir, experimentar i adquirir nous coneixements, aprofitant aquest camp. Inicialment amb una idea molt ambiciosa que, ràpidament, gràcies a l'experiència del director del projecte, es va anar adequant a les possibilitats reals.

Aquest projecte obre les portes a tota una línia de investigació posterior que possibilitaria l'augment de les prestacions i possibilitats d'uns vehicles aeris que volen sense pilot, de forma autònoma o comandats des de terra, executant missions en aplicacions reals.

2. OBJECTIUS

Dins de la quantitat i la complexitat dels subsistemes que componen un vehicle aeri no tripulat, ens hem centrat en només dos d'ells per a establir els objectius d'aquest projecte.

Més en concret, es dissenyarà i s'implementarà físicament la part corresponent al control des de terra d'un tipus concret d'aeronau. S'afegirà també un sistema autònom d'estabilització, mitjançant el qual, el vehicle aeri es mantindrà estable i horitzontal en els seus eixos longitudinal i transversal sense la intervenció del pilot en terra.

El projecte consistirà en una part d'integració d'equips electrònics ja existents, i una segona part amb el disseny del programari i maquinari necessaris per al seu conjunt funcionament.

Per tal de poder veure la funcionalitat del nostre disseny, s'utilitzaran altres "accessoris" que sense haver estat dissenyats estrictament dins del projecte, ens permetran interactuar amb el sistema. Parlem en concret del software de control de l'estació terra amb el PC, el propi vehicle, el joystick, l'equip de ràdio i els actuadors (servos).

Per al nostre cas, utilitzarem components ja habituals en aquest tipus d'aparells, intentant adaptar-los a les necessitats del projecte. D'aquesta forma s'integrarà tot un sistema complex, a partir de dispositius i maquinari, alguns d'ells ja existents i d'altres dissenyats a mida per a l'aplicació.

Des de un punt de vista més funcional, l'objectiu a assolir és el de, per una banda, governar des de terra una aeronau d'unes característiques concretes, de la mateixa forma que ho faria un aeromodelista, però amb la important diferència de que ho estem fent mitjançant un ordinador personal. Això deixa les portes obertes a posteriors aplicacions mitjançant les quals, el propi ordinador pot governar de forma automàtica i programada, un determinat tipus de vol (seguiment de "waypoints", vol en cercles, enlairament automàtic, etc...).

D'altra banda, precisament com a complement d'aquesta primera part, cal que el vehicle sigui en tot moment autoestable. Tenint en compte que el pilot no tindrà contacte visual, no podrà corregir els errors en la seva estabilitat. Encara més, de cara a pensar en un futur vol automàtic, cal que l'aeronau es mantingui estable de forma autònoma.

Doncs aquesta part d'estabilització, es pretén dissenyar també en aquest projecte. Mitjançant un sensor de moviment, concretament un acceleròmetre, el microcontrolador situat al vehicle haurà de llegir i corregir la seva posició, portant l'aeronau a una correcta posició de vol.

Com ja es pot entreveure, s'obren moltes noves possibilitats en el disseny d'aquestes màquines voladores, que hauran de quedar per a futurs projectes.

3. VANT (Vehicle Aeri no Tripulat)

Abans d'entrar en matèria directament, he vist oportú fer una petita introducció general sobre el camp en el que es desenvolupa aquest projecte.

Els UAV (Unmanned Aerial Vehicle) o VANT segons la nostra denominació, sorgeixen dins de l'àmbit militar durant el final del segle XX. Es tracta de vehicles dissenyats per a portar a terme missions especials, de cert risc o complexitat, però sense pilots ni tripulació a bord.



Figura 3-1. Strato Hawk System SRS – 4A (US)

No obstant, més recentment, en alguns països comencen a aparèixer empreses d'àmbit civil, que o bé comercialitzen plataformes per a diferents aplicacions, o bé ofereixen serveis utilitzant aquestes plataformes.



Figura 3-2. Aerovisión Fulmar (San Sebastián, ES.)

Es classifiquen segons diverses característiques com poden ser la seva missió, segons el seu sostre o l'abast màxim. Així en podem trobar:

- De **Reconeixement** , per a transmissió de dades militars
- **Investigació**, provant els sistemes en desenvolupament

- **Civils o comercials**

- **CLOSE** amb 1500 m d'altitud (5000 peus aprox.), fins a 10 Km d'abast.
- **HALE** (*High altitude, Long endurance*) fins als 9000 m (30000 peus aprox.) i abast indeterminat.



Figura 3-3. Schiebel S-100 Air Vehicle.

Tot i així, els costos actuals d'aquestes plataformes és encara lluny de ser assequibles. En molts casos, la manca de legislació, el desconeixement per part dels particulars, o els impediments legals, fan que aquesta tecnologia trobi certes dificultats alhora d'obrir-se pas en l'àmbit civil.

És per això, que apareixen altres línies de investigació i desenvolupament, de tipologia molt més modesta, de vegades barrejant-se amb el món del hobby, però no per això menys fructíferes. D'aquesta manera existeixen en el mercat diversos dispositius electrònics, diferents dissenys d'aeronaus, i una bona oferta de fabricants, que ofereixen totes aquestes solucions per tal d'integrar, a molt més baix cost, veritables i complets sistemes i plataformes d'aquest tipus de vehicles.

Tot depèn aleshores de les aplicacions per a les quals s'hagin de dissenyar. Com es pot avançar, no tindrà les mateixes característiques quant a tipus de vol, autonomia, etc, una aeronau que només hagi de fotografiar un indret concret, a una determinada alçada, que una altra que hagi de realitzar la vigilància aèria d'un indret geogràfic de determinades dimensions.

Entre les diferents aplicacions en les que poden actuar i de fet, en alguns casos ja actuen, existeixen les següents:

- Vigilància d'incendis forestals
- Seguiment i control de línies elèctriques
- Imatges i fotografia aèria
- Agricultura de precisió
- Teledetecció (SIG, LIDAR,...)
- Vigilància fronterera
- Recerca de bancs de pesca
- Suport en salvament i rescat



Figura 3-4. Exemple de Imatge aèria.

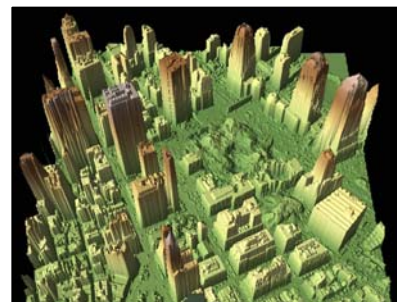


Figura 3-5. Model 3D Urbà, obtingut amb sistema LIDAR.

No hem d'oblidar que, pel fet de tractar-se de vehicles no tripulats, i en molts casos, de costos molt més reduïts en comparació amb els vehicles aeris convencionals, moltes d'aquestes aplicacions agafen encara més importància i valor afegit.

Per exemple, en alguna determinada missió de rescat es poden fer vols nocturns, augmentant les possibilitats d'èxit. Normalment els mitjans aeris no treballen en condicions nocturnes. En d'altres casos, el fet de no posar en perill la vida dels pilots, permet executar tipus de missions que d'altre forma serien inqüestionables.

Tot i així, almenys en el nostre país, existeix una manca de legislació que fa, si més no qüestionable, la utilització d'aquests mitjans. Cal aleshores una regulació i unes normatives que permetin a les empreses exercir amb absolutes garanties legals, fins ara només emmarcades dins el Codi d'Aviació Civil i la normalització de la modalitat esportiva de la qual en alguns casos en deriva.

4. AERODINÀMICA d'ESTABILITZACIÓ i CONTROL

L'aerodinàmica és una branca de la mecànica de fluids que estudia les accions que apareixen sobre els cossos sòlids quan existeix un moviment relatiu entre aquests i el fluid que els envolta.

Evidentment l'estudi d'aquesta branca de la ciència queda ben lluny, tant de l'objectiu d'aquest projecte com dels meus coneixements. No obstant, si el que volem és controlar i estabilitzar una aeronau encara que sigui a petita escala, ens serà imprescindible conèixer com hauran d'actuar els diferents controls per tal d'assolir els moviments i correccions necessàries.

El tipus d'aeronau sobre el que es basa el projecte, es correspon amb un tipus d'avió convencional. És important aquesta precisió, perquè en funció d'un model o altre de vehicle, les formes de control poden variar considerablement. En la figura 4-1, es pot veure quins són els controls mecànics elementals, sobre els que cal actuar per tal de produir variacions en la posició d'una aeronau d'aquestes característiques.

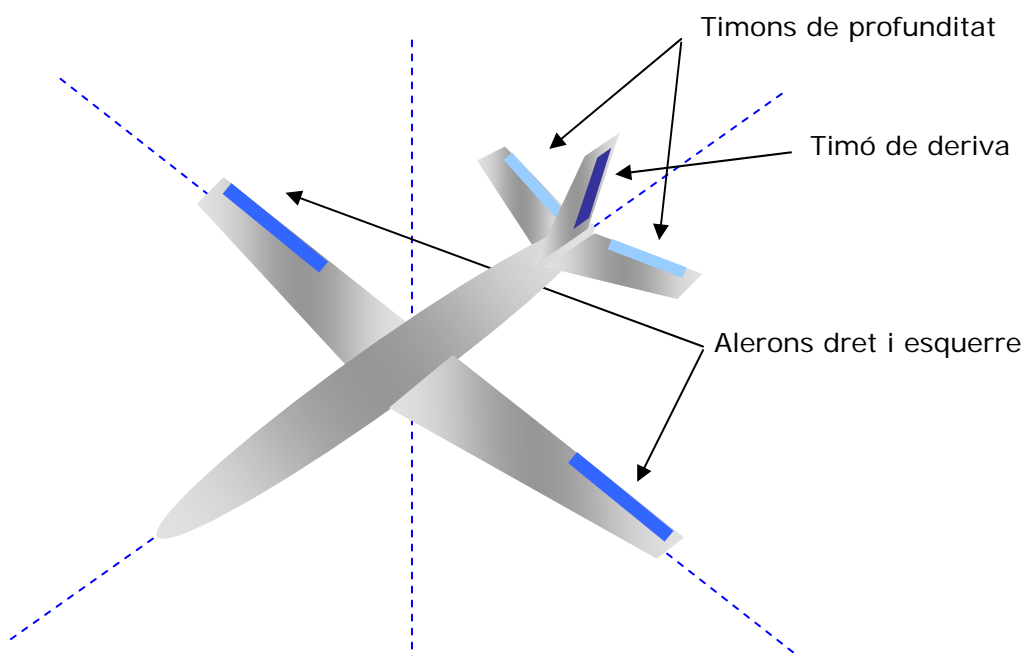


Figura 4-1. Elements mecànics, responsables del posicionament

En la realitat existeixen molts altres controls addicionals (flaps, flaperons, aerofrens, etc) per tal de modificar la sustentació de l'aeronau en moments concrets, com són l'enlairament i l'aterratge, i encara d'altres de propòsit més especial.

No obstant, per al nostre cas, ens centrarem únicament en els representats en la figura 4-1, donat que seran els únics sobre els que actuarà el nostre disseny.

Alerons

Aquests elements situats longitudinalment en els extrems anteriors de les ales, són els encarregats del moviment d'alaveig. Aquest moviment es produeix quan la resistència que produeix l'aire en contacte amb les ales, modifica la sustentació en forma inversa per a cada una d'elles. Aquesta diferència de sustentació és la que produeix la inclinació en l'eix longitudinal, utilitzada en el viratge del vehicle.

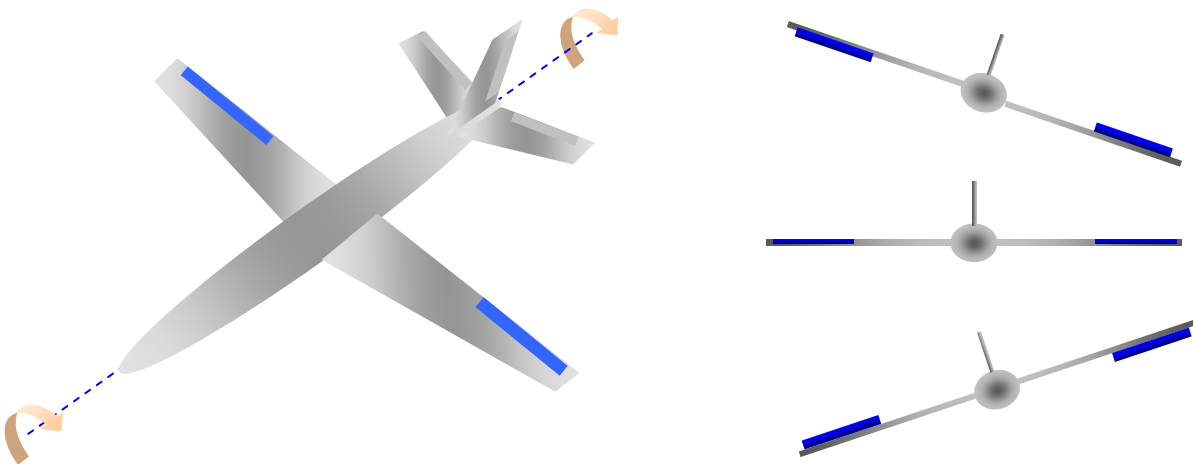


Figura 4-2. *Moviment i vista frontal de la inclinació longitudinal produïda pels alerons.*

En la figura 4-2, es pot veure com el moviment d'aquests controls és asimètric. És a dir, quan un aleró canvia la seva posició, el contrari es mou en sentit oposat, provocant la conseqüent variació en l'eix longitudinal de l'aeronau.

Com que aquesta inclinació és part de la estabilització que es busca, ja podem avançar que aquests controls no només seran modificats de forma manual pel pilot, sinó que necessitaran ser modificats de forma automàtica quan el vehicle hagi de autoestabilitzar-se.

Timons de Profunditat

De forma semblant als alerons, els timons o timó (en alguns casos es únic) de profunditat, es situen en la part anterior del pla horitzontal de l'estabilitzador de cua. En aquest cas, és el responsable de les variacions en la inclinació transversal de l'aeronau.

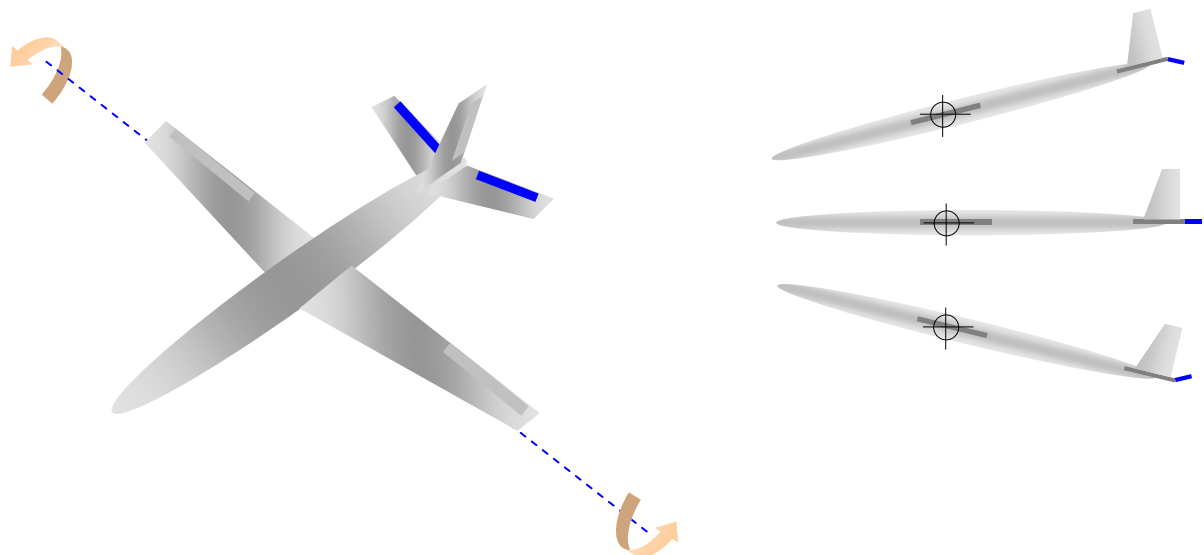


Figura 4-3. *Moviment i vista lateral de la inclinació produïda pels timons de profunditat.*

Novament, quan aquest control s'inclina, augmenta la resistència amb l'aire, alhora que augmenta o disminueix la sustentació en la part anterior de l'aeronau. D'aquesta forma varia la posició de la cua, de forma que es pot inclinar respecte del seu eix transversal.

A diferència dels alerons, aquest moviment mecànic és únic en tota la part mòbil dels timons. Hi hagi un o dos timons, en els casos dels empenatges horitzontals, es produirà un únic moviment en el mateix sentit.

De la mateixa manera que en el cas dels alerons, els timons de profunditat també tenen força responsabilitat pel que fa a l'estabilització del vehicle, en aquest cas però, respecte del seu eix transversal. Per aquest motiu, tant la part de control del pilot com la part automàtica d'estabilització, hauran d'actuar sobre ells.

Timó de deriva

És l'últim dels elements que es veuen afectats en la part de control relativa al nostre disseny. Es tracta en aquest cas d'un control mecànic, que en molts casos només actua com a complement dels alerons alhora de realitzar els viratges de l'aeronau.

Sigui com sigui, de cara al control i fins i tot de cara a futures aplicacions d'autopilotatge, hem de poder actuar sobre aquest element.

Es situa també en la cua de l'avió, però aquest cop en el pla vertical d'aquesta. És l'encarregat de les variacions sobre l'eix vertical del vehicle, però sense modificar el pla horitzontal d'aquest.

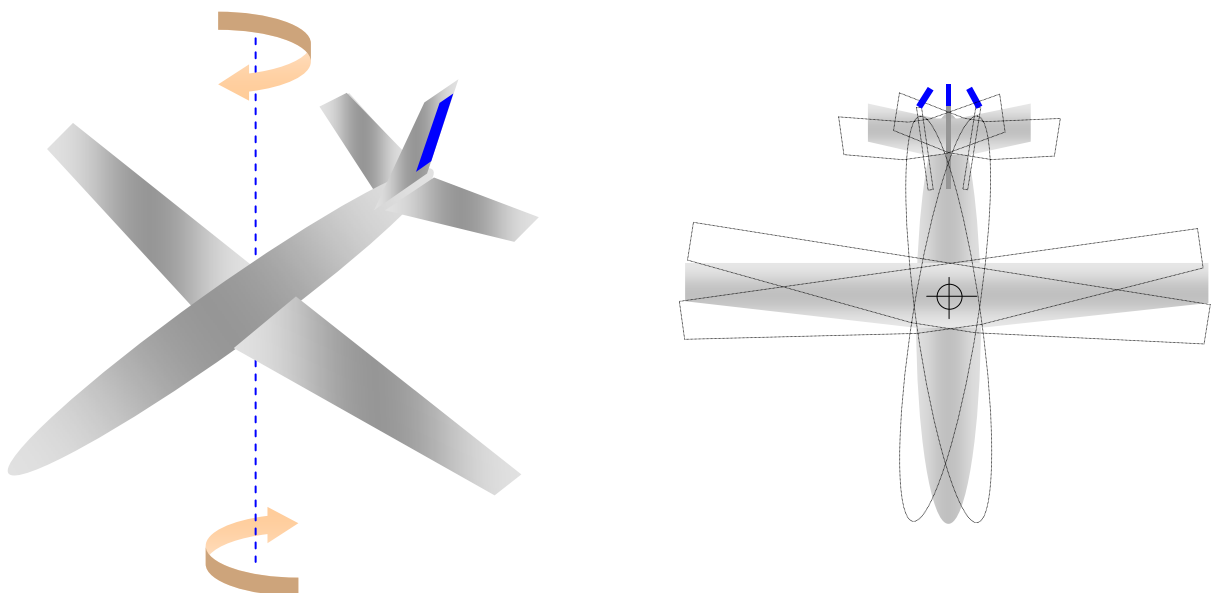


Figura 4-4. Moviment i vista superior de la modificació produïda pel timó de deriva.

Per tant, donat que aquest element no modifica el pla horitzontal, no es veurà implicat en la part d'estabilització automàtica. Únicament l'haurem de modificar en la part del control manual, per tal que el pilot pugui utilitzar-lo alhora de fer girs i per a la compensació de sustentació necessària durant el pilotatge.

4.1. ELEMENTS MECÀNICS I ELECTRÒNICS DE CONTROL

Com hem explicat anteriorment, aquests són els controls concrets que s'utilitzen en aquest tipus concret de vehicle aeri. És el cas del model utilitzat durant el projecte, motiu pel qual s'ha fet la descripció.

Per al moviment mecànic d'aquestes parts, s'utilitzen uns servomecanismes ja dissenyats per a tal propòsit. N'hi ha de molts tipus diferents en funció de l'aplicació per a la qual es destinen, amb diferent parell de força, mida, analògics o digitals, temps de resposta i un llarg etc.



Figura 4-5. Servo analògic
Hitec HS 65MG

Aquest tenen unes característiques de funcionament específiques, les quals haurem de conèixer i utilitzar per tal que realitzin els moviments com i quan el nostre disseny ho necessiti. El què es vol destacar, és que la majoria utilitzen el mateix tipus de "protocol" o senyal per a funcionar PWM (*Pulse Width Modulation*) concretament, per la qual cosa, el disseny realitzat en aquest projecte és fàcilment adaptable als diferents tipus de vehicles existents.

El motor

Tot i que no s'ha parlat, convé donar una noció bàsica del control d'aquest important element. Fins fa uns anys, els motors utilitzats en aquestes modalitats eren de combustió de mescla (glow, o oli amb benzina), molt semblants als utilitzats en petits ciclomotors o eines portables equipades amb aquests.



Figura 4-6. Motor de combustió OS de 20cc.

De fet encara avui dia, en les aplicacions en VANTs que requereixen de certa autonomia (superior a una hora de funcionament), encara són necessaris aquest tipus de dispositius mecànics, alimentats amb un tanc de combustible de mida concreta.

Doncs bé, l'element mecànic de control utilitzat per a modificar la potència subministrada pel motor, és igualment un servo com els vistos abans, que obren o tanquen el carburador oferint la quantitat de mescla requerida.

No obstant, amb les millores tecnològiques que ofereixen les bateries de **polímers de liti** (Li-Po), darrerament han proliferat i moltíssim els motors elèctrics, concretament els que no utilitzen escombretes anomenats *brushless*, augmentant així el seu rendiment.



Figura 4-7. Motor elèctric brushless, equivalent a un 70cc de benzina.

Per evitar modificacions en els sistemes de control en el pas dels anteriors motors de combustió als nous models elèctrics, apareix un nou element anomenat variador electrònic. Aquest nou element de control utilitza el mateix tipus de senyal PWM que s'utilitza en el moviment dels servos, de tal manera que no cal fer distincions ni nous dissenys alhora de governar, o bé un servo, o bé un motor tradicional de combustió, o bé un nou motor brushless.



Figura 4-8. Variador electrònic 30A de corrent nominal.

Això facilita molt les coses alhora d'implementar tota la part de control del nostre projecte.

Feta aquesta petita explicació, quedem-nos només amb la idea de quins controls han d'actuar per a provocar un determinat moviment de l'aeronau.

5. DESCRIPCIÓ GLOBAL del SISTEMA

Abans d'entrar en detalls tant del funcionament com dels diferents components que integren el sistema, es vol donar una visió global de tot el conjunt. Serà més endavant, en posteriors apartats quan s'entrarà en els detalls de cada un dels components, però per tal d'explicar els modes de funcionament convé tenir aquesta idea general.

A grans trets, el sistema es compon de dues parts fonamentals. Aquestes són:

- El vehicle aeri amb tots els elements de control necessaris
- Estació de terra

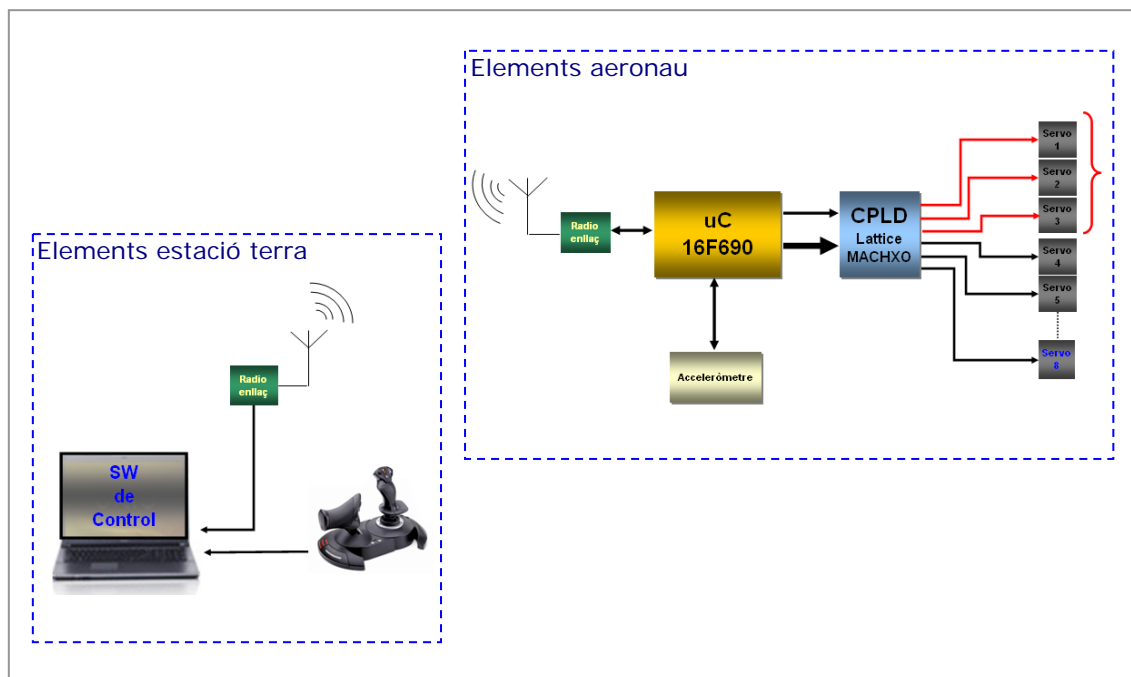


Figura 5-1. Diagrama de blocs del Sistema

El diagrama de blocs de la figura 5-1 ens mostra la configuració del sistema de forma molt bàsica. Pel que fa a l'estació de terra, els elements dels quals disposarà el pilot són l'ordinador personal amb el programari de control, el joystick per al pilotatge i el maquinari necessari per a la comunicació amb el vehicle.

Des del punt de vista del projecte, aquesta ha estat la part de menys elaboració, atès que no s'ha desenvolupat cap dels elements específicament per aquest. El programari de control juntament amb el joystick formen part d'un projecte paral·lel que ha estat utilitzat per tal d'interactuar amb la resta dels elements. Així doncs, únicament el maquinari necessari en la comunicació ha estat desenvolupat específicament dins el marc del present projecte.

Diferent és el cas de la part instal·lada dins l'aeronau. Podem dir que aquesta és l'àrea de desenvolupament estrictament d'aquest projecte. És a dir, s'ha adaptat tot un maquinari específic, el qual inclou comunicacions, microcontrolador, CPLD, sensor (acceleròmetre) i actuadors (servos), per a funcionar conjuntament amb l'equip de l'estació de terra.

El propòsit inicial ha estat el d'utilitzar components econòmics, però que alhora poguessin complir les mateixes especificacions o expectatives de funcionament que tot un sistema de més altes prestacions i/o qualitat. És a dir, que si en un futur es vol ampliar l'abast de ràdio, el nombre o tipologia de sensors, afegir prestacions extremes, etc, el disseny del projecte permeti adaptar fàcilment els nous components, i mantenir els protocols de comunicacions així com la part fonamental del programari del microcontrolador.

Per tant, aquesta darrera part serà la que ens ocuparà força extensament durant la resta del text.

6. MODES DE FUNCIONAMENT

Un cop estan emmarcats els diferents elements, podem ja parlar del funcionament des d'un punt de vista d'usuari o de pilot.

El sistema es troba en constant comunicació base terra – aeronau. D'aquesta forma, existeix la possibilitat de governar l'aparell des de terra, responent aquest a les accions de pilotatge.

Per a fer el control de l'aeronau, el disseny implementa un protocol de comunicació de 8 canals, mitjançant el qual s'assigna per a cada canal un control mecànic en funció del tipus de control necessari en el vehicle.

No hem de confondre aquests canals amb diferents assignacions de freqüències. La freqüència de ràdio de treball és una única i concreta. Per canals en aquest cas, volem dir només diferents assignacions de dades que ens donarà el protocol de comunicacions, el qual es veurà més endavant. Veiem un exemple:

Canal de comunicacions	Assignació al receptor
Canal 1	Motor
Canal 2	Timó Deriva
Canal 3	Aleró dret
Canal 4	Aleró esquerre
Canal 5	Timó Profunditat
Canal 6	lliure
Canal 7	lliure
Canal 8	Control estabilització

Taula 6-1. Exemple d'assignació de canals.

Quan es parla de control manual, correspon al pilot tant la part de control general, com d'estabilització. El sistema d'estabilització automàtica és activat manualment des de terra, a voluntat del pilot.

Es poden preveure funcionalitats extres com activació automàtica de l'estabilització en cas de pèrdua de senyal, o també en cas de no rebre una variació d'ordres actuant sobre els controls d'estabilització.

No obstant, aquestes darreres opcions no han estat implementades i queden per a futurs projectes.

6.1. CONTROL MANUAL

El control manual es realitza des de l'estació de terra amb l'ajuda dels elements:

- Ordinador personal amb programari de control
- Maquinari de comunicacions
- Joystick de control

Funciona pràcticament com es faria el pilotatge d'un avió en un videojoc. El pilot té accés en temps real a tots els controls disponibles a bord de l'aeronau i aquests responen segons la voluntat del pilot.

Fins aquí es pot assimilar el sistema a un control remot tradicional, en aquest cas canviant per complert el protocol de comunicacions i inserint un ordinador personal que, en un futur pot afegir moltes aplicacions extres, com poden ser autopilotatge de "waypoints", vols programats (circulars o delimitats), enlairament automàtic, etc.

En aquest mode de funcionament, el maquinari a bord, rep per ràdio les ordres directes del pilot, i sense cap processament extra (només detecció d'errors), són enviades directament al maquinari encarregat de traduir aquestes ordres en el senyal necessari per a actuar sobre tots els controls mecànics de l'aeronau. Es veurà posteriorment amb molt més detall.

El programari de control, que volem insistir es troba fora de les activitats realitzades en aquest projecte, permet establir les condicions i especificacions per a cada tipus de control, segons l'eix del joystick desitjat, recorreguts màxims i mínims dels servos, sentit de gir, etc.

D'aquesta forma, queda definit tot un sistema més o menys complex, mitjançant el qual es pot governar el vehicle de forma totalment manual, sempre i quan hi hagi comunicació establerta.

Cal incidir en aquest apartat, en que l'aeronau només disposa de l'estabilització que li pugui donar el pilot. És a dir, que la funció automàtica d'estabilització no funciona en aquest mode fins que és activada manualment des de terra.

6.2. ESTABILITZACIÓ AUTÒNOMA

Aquesta funcionalitat és activada des de terra pel pilot, mitjançant un dels pulsadors del joystick de control. Quan és activa, els controls associats al moviment en el pla horitzontal, ja no poden ser intervinguts pel pilot, quedant governats estrictament pel maquinari de l'aeronau.

En aquest mode de funcionament l'aeronau treballa de forma combinada, utilitzant les ordres d'estabilització generades des del microcontrolador a bord, i les provinents de terra, en aquest cas dels canals "lliures" disponibles per al pilot.

Per al funcionament, que veurem detallat en els posteriors apartats, el maquinari a bord utilitza un sensor de posicionament (acceleròmetre) que llegeix en tot moment la posició en els dos eixos horitzontals de l'aeronau. El microcontrolador també a bord, processa aquesta informació i envia les ordres de correcció de posició als controls mecànics implicats, és a dir, els alerons dret i esquerre, i el timó de profunditat.

La resta de controls mecànics, el timó de deriva i la potència del motor, són enviats directament segons les ordres rebudes des de terra, sense cap processat extra, tret de la detecció d'errors.

En aquest mode, el pilot pot fer un control manual, preocupant-se només de girar l'aeronau amb el timó de deriva, afegint o no potència del motor en funció de les necessitats del vol.

La resta de controls són actuats de forma automàtica, per tal de mantenir sempre l'aeronau en una posició estable i horitzontal.

Aquest mode de funcionament pot ser molt útil quan el pilot no disposa de contacte visual amb el vehicle, podent així “despreocupar-se” de l'estabilització del mateix.

A nivell de taula podem veure com queden els orígens de control dels diferents mecanismes:

Control mecànic	Origen de les ordres	
	Mode control manual	Mode Estabilització automàtica
Motor	Manual pel pilot	Manual pel pilot
Timó de deriva	Manual pel pilot	Manual pel pilot
Alerons drets i esquerre	Manual pel pilot	Automàtic, pel uC a bord
Timó de profunditat	Manual pel pilot	Automàtic, pel uC a bord

Taula 6-2. Taula amb l'origen de les ordres de control en funció del mode de funcionament.

7. DESCRIPCIÓ dels COMPONENTS del SISTEMA

Abans s'ha vist de forma molt general els diferents components del sistema complert. Entrem ara en detall explicant cadascun d'ells, amb la seva funció i característiques.

7.1. ELEMENTS de l' ESTACIÓ TERRA

A nivell de l'equipament necessari en l'estació de terra, comentem que només s'ha fet una adaptació dels elements necessaris per a interactuar amb el nostre projecte. S'ha partit d'elements ja dissenyats per tal d'integrar-los convenientment per a ser utilitzats. No obstant, cal comentar-los.



Figura 7-1. Imatge dels components de l'estació de terra.

7.1.1. Ordinador personal

Poca cosa a comentar, tret d'un equip portàtil (només per comoditat), amb una capacitat de processador actual i connexió USB per tal de connectar tant el joystick per al control com el convertidor USB-TTL que ens emularà un port de comunicacions sèrie.

En el nostre cas estem treballant amb S.O. Windows XP. Aprofitem aquest apartat per explicar que, per a una tasca de control com la del projecte, és recomanable treballar amb un sistema que permeti l'execució de tasques a temps real.

Per tal que el control sigui prou robust, alhora que fiable, es necessita d'aquesta funcionalitat, així com poder donar la màxima prioritat a la aplicació de control. Podem imaginar-nos el que pot passar si el pilot perd el control de l'aeronau per culpa de qualsevol interrupció deguda a un esdeveniment extern i inesperat.

Doncs bé, remarquem que el programari utilitzat, amb la configuració de tot el conjunt aplicació i sistema operatiu, en el nostre cas concret, no treballa correctament en temps real com caldria esperar. Més endavant es veurà que ens ha comportat alguna limitació, com la quantitat de trames per segon que poden ser enviades al vehicle. Això no seria recomanable en un sistema real, però ens ha servit de base per a poder verificar el propòsit del nostre projecte.

7.1.2. Palanca de comandament (Joystick)

Per a la nostra aplicació s'està utilitzant un joystick de jocs convencional de la firma Logitech.

El que ens interessa a nosaltres és que conté les possibilitats necessàries per al pilotatge de qualsevol tipus d'aeronau. Per fer-ho disposa de 4 eixos de moviment continu i proporcional, així com una quantitat de pulsadors que ens poden ajudar en el control. Per exemple per a l'activació del mode d'estabilització automàtica, s'utilitzen dos pulsadors laterals.



Figura 7-2. Joystick Logitech EXTREME 3D PRO utilitzat en el projecte.

La connexió amb l'ordinador personal es realitza mitjançant USB. L'aplicació de control, programada en VisualBasic.net incorpora una llibreria per tal de llegir les dades provinents del joystick. No comentarem aquesta part de l'aplicació, per trobar-se fora de l'abast del projecte, com ja havíem comentat.

El que sí que expliquem és el tipus de dades rebudes. Per a cada eix, el programari llegeix un valor de 16 bits (de 0 a 65535).

Aquest és processat pel programa de control, de tal manera que aplica les modificacions necessàries per a cada canal.

Per al cas dels pulsadors, també s'assigna un valor diferent per tal de llegir quin és el pulsador premut i d'aquesta forma, assignar el valor que necessitem en aquell canal per a una determinada funció.

7.1.3. Programari de control de l'estació terra

Tot i que aquesta aplicació de control no ha estat desenvolupada durant el present projecte, sinó que forma part d'un projecte anterior, cal que comentem mínimament el seu funcionament. No obstant, sí que ha calgut adaptar les seves característiques a les necessitats del nostre projecte.

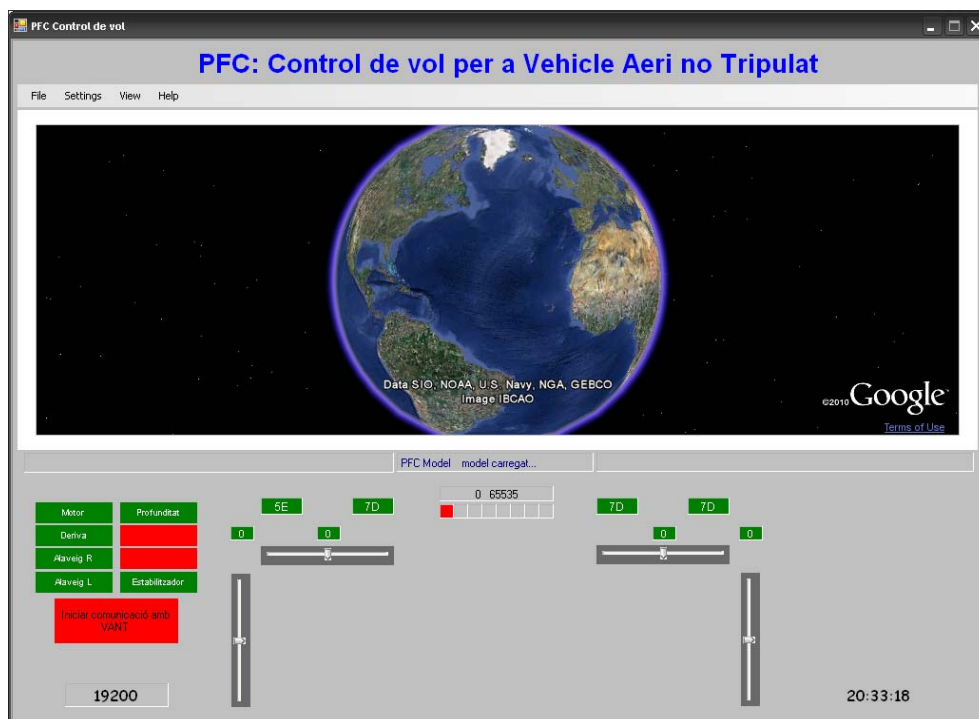


Figura 7-3. Vista principal del programa de control

De forma resumida, aquesta petita aplicació de control, s'encarrega de llegir les dades del joystick, realitza les adaptacions necessàries en funció de la configuració requerida, defineix una trama amb les dades de posició de cada canal, afegeix un checksum de seguretat, i finalment, aquesta trama és enviada a través del port sèrie de l'ordinador.

Aquesta lectura del joystick, processat de trama i enviament de trama, és realitzada teòricament en intervals de 50 ms. Diem teòricament perquè això és el que es pretén amb la funció programada en VisualBasic.net, però desafortunadament, aquest interval de temps ens ve limitat per altres factors, més relacionats amb la nostra configuració del sistema operatiu. Per aquest motiu, no s'està treballant en temps real com caldria fer.

La informació o configuracions de cada canal poden ajustar-se per a cada model en concret i tenir així diferents arxius de configuració en funció del tipus de vehicle que s'hagi de pilotar. Es poden activar o desactivar canals si es desitja, fer un ajust fi de cada canal per a les correccions de posició del vehicle, veure les dades que es recullen del joystick, etc.

El que és important és introduir com s'ha definit la trama de comunicacions sobre el protocol RS-232.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13
FF	FE	FD	Pos Canal 1	Pos Canal 2	Pos Canal 3	Pos Canal 4	Pos Canal 5	Pos Canal 6	Pos Canal 7	Pos Canal 8	CHK 1	CHK 2
Sincronització de trama			Dades de posició de cada canal								Checksum calculat	

La taula 7-1 ens mostra una trama complerta. Com que es treballa sobre RS-232, aquesta seria equivalent a enviar un conjunt de 13 caràcters, de la qual nosaltres ens quedem amb el valor del contingut de cada Byte, enlloc de la seva representació ASCII.

Taula 7-1. Trama de comunicacions enviada des de la base a l'aeronau.

El programari s'ha adaptat per tal de generar correctament aquesta trama i així poder ser interpretada des del microcontrolador del vehicle. S'ha afegit dades per a la sincronització i dades de checksum per a la detecció d'errors. El detall serà explicat en el posterior apartat 8.4.

De cara a futures aplicacions o projectes relacionats, podria ser força interessant basar-se en aquesta aplicació, afegint les prestacions de vol automàtic, seguiment del vol a través de telemetria, etc.

Això sí, caldria tenir ben en compte la situació actual i millorar en el possible el treball en temps real.

7.1.4. Maquinari de l'estació terra

El disseny de menys dificultats de desenvolupament, pel que fa a maquinari de tot el projecte, es troba precisament en l'estació terra. La seva funció principal és la d'enviar via ràdio la trama generada des de l'aplicació de control en l'ordinador, cap a l'aeronau.

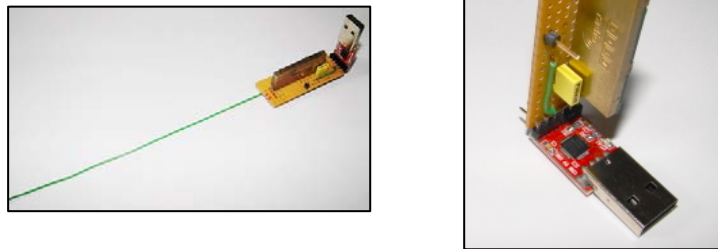


Figura 7-4. *Fotografies del conjunt muntat en una única placa*

Per al seu disseny, s'han fet servir dos mòduls que es poden trobar molt fàcilment al mercat, amb un cost força econòmic. Es tracta del transceptor de ràdio, anomenat de vegades radiomòdem i l'adaptador descrit com a USB-TTL.

La majoria dels més recents ordinadors portàtils ja no incorporen el port sèrie conegut com RS-232 (DB9). Aquest ha estat substituït majoritàriament pel USB. Per aquest motiu, es troben al mercat una bona gamma de dissenys dedicats a la conversió de la informació del port USB, cap a dades compatibles amb els estàndards RS-232 que encara han de ser utilitzats per molts perifèrics.

No obstant això, encara més, la majoria de dispositius treballen amb rangs de tensió compatibles amb tecnologia TTL (0 a 5 v.), a diferència de l'estàndard RS-232, que treballava dins un rang de tensions de -12 a 12 v.

Doncs d'aquesta conversió de tensions també s'encarrega aquest element.

Un cop hem adaptat els senyals a un nou protocol, basat en el RS-232, però amb nivells de tensió TTL, aquests senyals ja poden ser enviats a través del radiomòdem.

Per tot això, expliquem a continuació amb una mica més de detall aquests dos elements.

Adaptador USB – TTL

D'aquest singular mòdul, se'n poden trobar moltes varietats. N'hi ha d'inserits directament en els connectors instal·lats en un cable, muntats en placa, etc.



Figura 7-5. Vista de l'adaptador USB-TTL utilitzat.

El que tenen en comú la majoria d'ells és la utilització de l'integrat *FTDI chip* un convertidor **USB slave** (esclau) força estandaritzat i utilitzat. Tant és així que hi ha diversos fabricants treballant aquest tipus de controladors.

En el nostre cas, s'ha fet servir el model CP2102 de la firma SILICON LABS.

Adjuntem algunes dades, gentilesa del fabricant, només com a referència.

Single-Chip USB to UART Data Transfer

- Integrated USB transceiver; no external resistors required
- Integrated clock; no external crystal required
- Integrated 1024-Byte EEPROM for vendor ID, product ID, serial number, power descriptor, release number, and product description strings
- On-chip power-on reset circuit
- On-chip voltage regulator: 3.3 V output
- 100% pin and software compatible with CP2101

USB Function Controller

- USB Specification 2.0 compliant; full-speed (12 Mbps)
- USB suspend states supported via SUSPEND pins

Asynchronous Serial Data BUS (UART)

- All handshaking and modem interface signals
- Data formats supported:
 - Data bits: 5, 6, 7, and 8
 - Stop bits: 1, 1.5, and 2
 - Parity: odd, even, mark, space, no parity
- Baud rates: 300 bps to 1 Mbps
- 576 Byte receive buffer; 640 byte transmit buffer
- Hardware or X-On/X-Off handshaking supported
- Event character support
- Line break transmission

Virtual COM Port Device Drivers

- Works with Existing COM Port PC Applications
- Royalty-Free Distribution License
- Windows Vista/XP/Server 2003/2000
- Mac OS-X/OS-9
- Linux

USBXpress™ Direct Driver Support

- Royalty-Free Distribution License
- Windows Vista/XP/Server 2003/2000
- Windows CE

Example Applications

- Upgrade of RS-232 legacy devices to USB
- Cellular phone USB interface cable
- PDA USB interface cable
- USB to RS-232 serial adapter

Supply Voltage

- Self-powered: 3.0 to 3.6 V
- USB bus powered: 4.0 to 5.25 V

Package

- Pb-free 28-pin QFN (5x5 mm)

Ordering Part Number

- CP2102-GM

Temperature Range: -40 to +85 °C

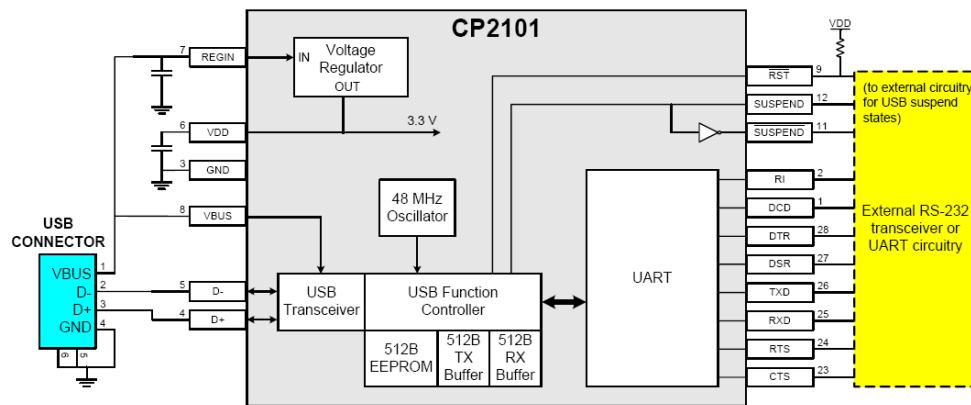


Figura 7-6. Diagrama de blocs del controlador USB slave CP2102

Transmissor Ràdio

Aquest mòdul té la particularitat de ser utilitzat tant com a Transmissor com a Receptor. D'aquí el nom de transceptor. Per aquest motiu, li dediquem un apartat 7.2. on explicarem el detall, parlant tant de la part receptora com transmissora.

Esquema muntatge

L'esquema general del muntatge on podem apreciar la seva simplicitat, es pot veure en las següent figura 7-7:

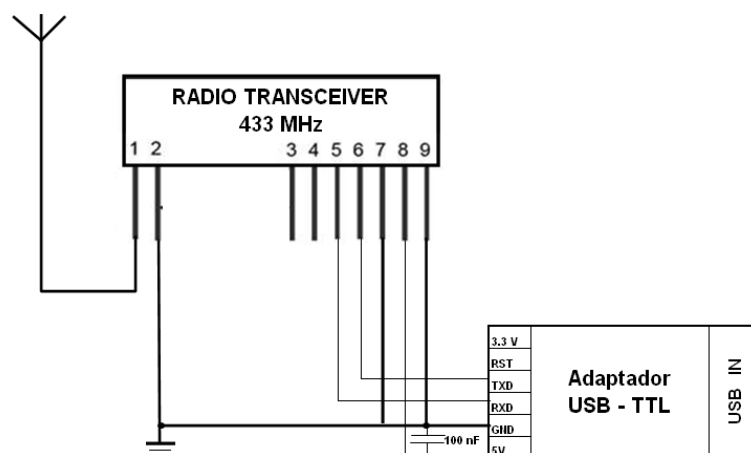


Figura 7-7. Esquema del muntatge utilitzat en el maquinari de l'estació base.

7.2. TRANSCEPTORS del SENYAL de RÀDIO

Per a la transmissió per ràdio, hem volgut també utilitzar elements econòmics. Amb el mateix tipus de protocol de comunicacions, augmentar l'abast o d'altres prestacions, no hauria de suposar cap problema, sinó només la substitució d'aquests mòduls.

Aquests en concret tenen la particularitat de ser força senzills d'utilitzar, aconseguint la transmissió de dades en un rang aproximat de 250 m en visió directa, segons fabricant. Evidentment no seria distància suficient per a cap aplicació VANT, però sí que ho és per a la verificació i el desenvolupament d'aquest projecte.

Hem utilitzat doncs, la parella de models ER400TS de la firma **easyRADIO**. Es troben disponibles en la banda de 433 - 434MHz i 869.85Mhz & 902 - 928MHz. Nosaltres fem servir la banda de 433MHz, d'acord amb la legislació del nostre país.

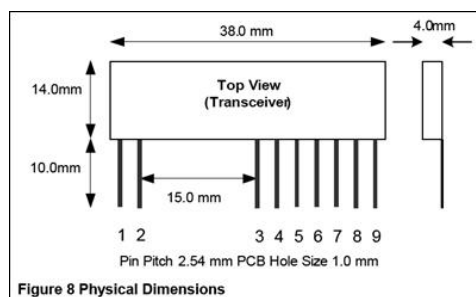
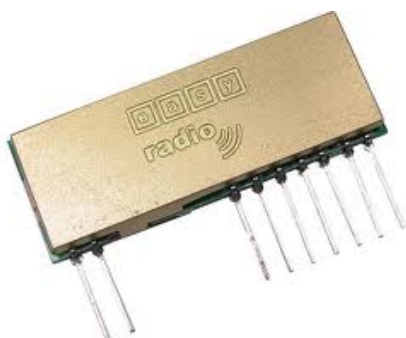


Figura 7-8. easyRADIO , ER400TS transceiver, utilitzat en el nostre muntatge.

Pin Description			
Pin No	Name	Description	Notes
1	Antenna	50Ω RF input/output. Connect to suitable antenna.	
2	RF Ground	RF ground. Connect to antenna ground (coaxial cable screen braid) and local ground plane. Internally connected to other Ground pins.	
3	RSSI	Received Signal Strength Indication	
4	Busy Output	Digital Output to indicate that transceiver is ready to receive serial data from host.	CTS function
5	Serial Data Out	Digital output for received data to host	
6	Serial Data In	Digital input for serial data to be transmitted	
7	Host Ready Input	Digital Input to indicate that Host is Ready to receive serial data from transceiver	RTS function
8	Vcc	Positive supply pin. +2.5 to +5.5 Volts. This should be a 'clean' noise free supply with less than 25mV of ripple.	
9	Ground	Connect to supply 0 Volt and ground plane	

Taula 7-2. Descripció de pins del transceptor.

Com a característiques generals hem de destacar les següents:

- Síntesi de freqüència controlada per cristall
- Sensibilitat del receptor 130dBm @ 19.2 Kbps
- Fins a 10 mW de potència de transmissió @ 433MHz
- Voltatge de treball 2,5v ~ 5,5v
- Consum de corrent al voltant dels 25 mA, amb opció de stby amb 120uA
- Opcions programables per usuari:
 - Freqüència de treball
 - Baud rate (velocitat de transmissió)
 - Potència de sortida

Incorpora un microprocessador intern mitjançant el qual implementa funcions addicionals com són control de paritat, programació a mida per l'usuari, selecció digital del canal a utilitzar dins la banda de transmissió, CRC per al control d'errors, etc.

Totes aquestes característiques fan que sigui un element molt versàtil, alhora que senzill d'utilitzar en moltes de les aplicacions. Mitjançant un programari addicional del fabricant, es té accés a moltes de les seves funcions, per tal de treballar i verificar el disseny durant el desenvolupament.

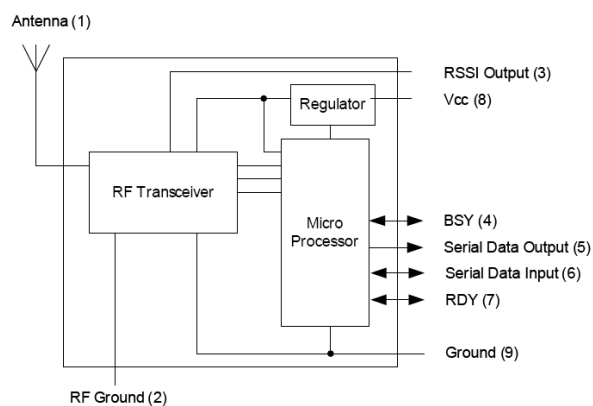


Figura 7-9. Diagrama de bloc intern del ER400TS.

El fabricant ens mostra un exemple típic d'aplicació, que es correspon amb el que s'ha desenvolupat. En el nostre cas, el Host (A) podria ser l'ordinador personal de l'estació de terra, i el Host (B) seria el microcontrolador situat al vehicle.

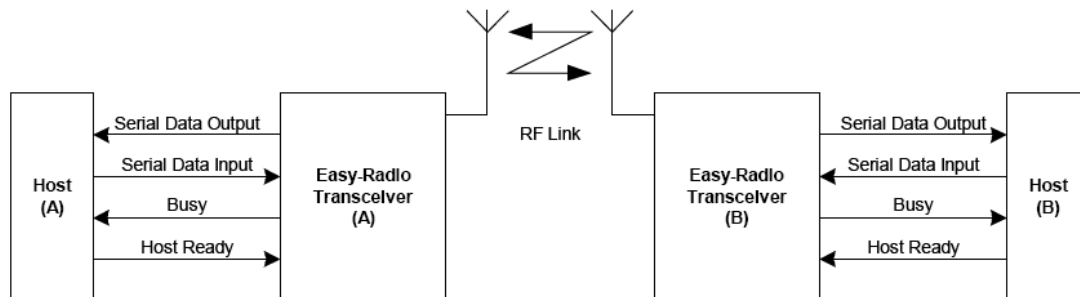


Figura 7-10. Exemple d'aplicació dels mòduls ER400TS.

A causa d'alguna limitació en el nostre maquinari, tant en el mòdul A com en el B, només hem utilitzat, les línies TX i RX per a la connexió de dades. Per exemple, en la connexió amb l'ordinador personal, a través de l'adaptador USB-TTL, no ens és possible accedir a pins extres com el *Busy* o el *Host Ready*. No obstant, això no ens implica cap limitació, donat que el control de la comunicació es pot fer a través del propi protocol.

En els dos casos ens hem connectat a un port sèrie. Un en el cas del microcontrolador de bord, que veurem més endavant, i l'altre a través de l'adaptador USB-TTL cap a l'ordinador personal.

Un fet a explicar és: per què utilitzar transceïdors, en lloc d'un transmissor i un receptor?

En qualsevol aplicació VANT que tingui prestacions més complertes, com el GPS o la navegació automàtica, es necessita de contínua comunicació entre la nau i terra i a la inversa. Aquesta prestació de comunicació bidireccional permet, per exemple, l'enviament de dades de telemetria des de l'aeronau cap al terra. Entre aquestes dades hi poden haver la posició donada pel GPS, l'alçada, la velocitat, el nivell de les bateries, la intensitat del senyal de ràdio, etc.

Totes aquestes incorporacions han de quedar també per a futurs projectes, però si més no, a nivell de comunicació ja s'havien contemplat durant les fases inicials d'aquest projecte.

La comunicació s'ha fet a una velocitat de transmissió de 19200 bps, per utilitzar un valor força estàndard tant en l'ordinador personal com en el microcontrolador. Aquesta velocitat de transmissió és més que suficient en l'aplicació. Podem veure-ho en un simple càlcul:

$$\text{Longitud de trama} = 13 \text{ Bytes} \cdot 10 \text{ bits} = 130 \text{ bits}$$

$$19200 \times \frac{1}{130 \text{ bits}} = 147 \text{ trames per segon}$$

Com es veu, es podrien arribar a tenir 147 trames cada segon, les quals són més que suficients per al pilotatge. No obstant, això ens vindrà limitat per una altra banda.

7.3. ELEMENTS del VEHICLE

Podem dir que tot el que s'ha desenvolupat fins ara, correspon a un 10 per cent del total del projecte. A on realment hi ha hagut més feina de desenvolupament ha estat sens dubte en tot el referent al vehicle.

Després d'haver vist en el diagrama de blocs de la figura 5-1, de què està composta aquesta part, entrarem ara en el detall de cada un dels elements.

Quant al disseny, s'ha tingut en compte les característiques necessàries per poder anar instal·lat en una aeronau, com són la seva mida i sobretot el pes. No obstant, queda encara molt per millorar, com per exemple la integració total de tot el maquinari en una sola placa.

Tot i així, s'ha mirat de compactar al màxim dins les possibilitats amb les que s'ha pogut treballar.

Veiem una imatge del conjunt de tot el maquinari:

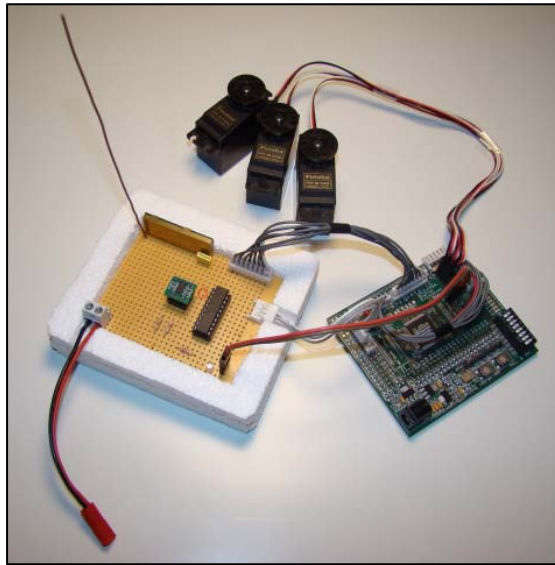


Figura 7-11a. Imatge del conjunt de maquinari instal·lat a l'aeronau.

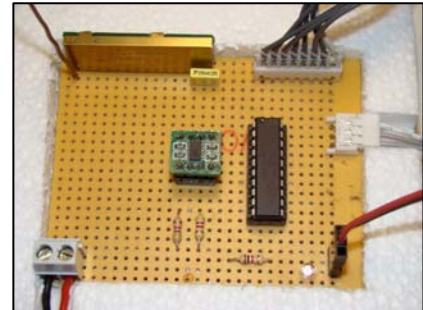


Figura 7-11b. Imatge del Disseny final de la PCB.

7.3.1. Actuadors (Servos)

Afortunadament per a nosaltres, per als moviments mecànics existeixen unes petites màquines, que ja es van introduir en l'apartat 4-1.

En aquest cas però, no parlem de les seves característiques generals, sinó del seu funcionament, i del mode en què nosaltres el farem treballar.

Tant si els servos són digitals, com si són analògics, tant si poden desenvolupar 1,5 Kg o 5 Kg de força, com si tenen un temps de resposta de 0,6s o de 0,08s, el seu principi de funcionament és molt semblant.

Es tracta d'un petit motor DC, muntat juntament amb uns engranatges que serveixen per a realitzar reduccions de velocitat angular i augmentar així el seu parell de força. Addicionalment, compten també amb un circuit per tal d'adaptar els senyals rebuts dels receptors per tal d'actuar correctament sobre el motor.



Figura 7-12. Vista d'un servo desmuntat amb les seues components.

Tret del seu funcionament intern, el que realment ens interessa a nosaltres, és saber quin és el senyal que nosaltres li hem d'aplicar per tal que actuï com esperem que ho faci.

A banda de com generem aquest senyal, que s'explicarà en el capítol 7.3.5. ens interessa més conèixer quin és aquest tipus de senyal i com varia la resposta del servo en funció de l'entrada que li apliquem.

Aquests elements disposen d'un connector amb tres cables, per a realitzar tot el control.



Figura 7-13. Connexions d'entrada utilitzades en tots els servos i variadors electrònics.

D'una banda hi ha l'alimentació comú en tots ells i en tot el circuit. De l'altra banda, tenim el senyal d'entrada PWM, que serà el senyal de control del posicionament del seu braç.

Aquest senyal, que li generarem a través del CPLD, ha de tenir les característiques que es mostren a la figura 7-14.

Un senyal PWM està format per un pols que es repeteix amb període fix amb un interval T_{on} , que pot variar però sense alterar el període total. És a dir, nosaltres hem de poder variar el T_{on} , mantenint sempre fix el període total del senyal.

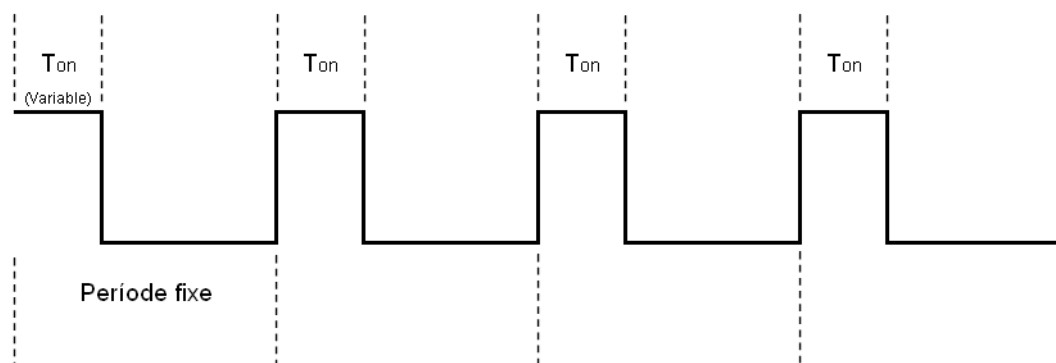


Figura 7-14. Senyal PWM (Pulse Width Modulation)

En el nostre cas, aquest període total és de 20 ms, el que equival a una freqüència de 50Hz.

En funció de la durada del T_{on} , que ve determinada pel fabricant de cada servo, aquest es posicionarà en un angle de gir concret. Normalment, els servos utilitzats en aquestes aplicacions, tenen un angle total de gir de 180° aproximadament.

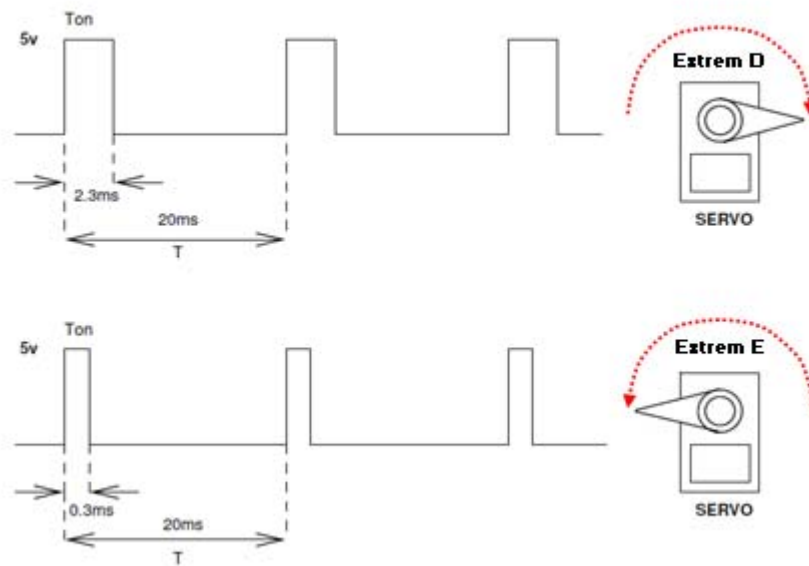


Figura 7-15. Angle de posicionament del servo, en funció de l'amplada del T_{on} .

Com ens mostra la figura 7-15, per aquest fabricant en concret, les amplades dels T_{on} , varien de 0,3 ms pel màxim a l'esquerra, fins a 2,3 ms pel màxim a la dreta. Tot això dins d'un període invariable de 20 ms. Afortunadament, la majoria de fabricants utilitzen valors i freqüències molt semblants per la qual cosa la majoria de servos s'adapten prou bé a la mateixa aplicació.

Doncs vist això, el següent pas serà generar aquest senyal tantes vegades com el necessitem, en funció dels canals i/o servos que s'hagin de connectar. En el nostre cas en farem servir 8 canals, per tant s'hauran de generar 8 senyals PWM diferents, ajustant-se a la necessitat de cada canal.

7.3.2. CPLD – FPGA

Per comoditat en el desenvolupament es va decidir utilitzar la placa d'avaluació de la firma Lattice, *MachXO Starter Evaluation Board*.

Tant per cost com per mida, així com el component que incorpora (MACHXO LCMX0256C), compleixen de sobres les necessitats del nostre disseny.



Figura 7-16. Placa de desenvolupament amb el MachXO LCMX0256 de Lattice.

Porta incorporat la CPLD o FPGA MachXO LCMX0256. Es tracta d'un integrat que compleix els requeriments d'aplicacions tradicionalment adreçades als CPLDs i FPGAs de baixa capacitat, tal i com ho descriu el fabricant.

Per tant, un element de maquinari programable que reuneix més que sobradament les característiques necessàries per a encabir el nostre circuit digital, on necessitem almenys de 8 comptadors de 11 bits, 8 comparadors de 8 bits , 8 registres, 8 portes NOR i elements addicionals per a generar el senyal de rellotge de 102,4KHz, entre d'altres.

Com a característiques principals resumim les següents:

- 256 LUT4s (Look Up Tables)
- Màxim nombre de I/O: 78
- Voltatge d'alimentació: 1.2/1.8/2.5/3.3v.
- No necessita memòria externa (incorporada de 2Kbits)
- JTAG port per a la programació

Pel que fa als elements de la placa d'avaluació, en destaquem:

- 8 LEDs per a propòsit general
- Switch DIP de 8 interruptors
- Alimentació de 5 vols regulats
- Oscil·lador de 33 MHz en placa
- Àrea per a prototipatge
- Port JTAG accessible per a la programació
- Accés a 81 pins d'E/S d'usuari

S'ha aprofitat l'àrea de prototipatge per a situar els connectors a on es connectaran els diferents servos de l'aplicació. La mateixa placa també permet afegir connectors de forma es pugui accedir fàcilment a les diferents E/S.



Figura 7-17. Imatge amb l'àrea utilitzat per al prototipatge.

7.3.3. Microcontrolador

Abans d'explicar la programació de la FPGA, necessitem haver vist el microcontrolador per entendre millor d'on apareixen certs senyals.

L'elecció ens ve donada de nou per utilitzar un component econòmic, però potser més important per treballar i aprofundir en un dels elements estudiats durant la carrera. Per aquest motiu s'ha optat per un model de la firma MICROCHIP, en concret el **PIC 16F690**.



Figura 7-18. PIC 16F690 encapsulat DIP utilitzat en el projecte.

La família de microcontroladors PIC és molt extensa i cal fer abans un bon anàlisi dels diferents models per tal de donar amb el que compleixi les necessitats però sense tenir un excés de recursos que no s'utilitzin.

Els punts que es van tenir en compte en l'elecció són:

- Oscil·lador intern, per la qual cosa ens estalviem dissenyar un circuit extern per al senyal de rellotge.
- Memòria de Programa de 4Kx14 i dades 256B (SRAM) + 256B(EEPROM).
- Quantitat de port d'E/S suficients: Port A (6 pins), Port B (4 pins) i Port C(8 pins).
- Mòdul intern RS-232, que s'utilitza per a la comunicació directa amb el transceptor (receptor) de ràdio.
- Mòdul intern I²C que s'utilitza per a la lectura de sensors externs, en el nostre cas, l'acceleròmetre.
- Opció de treballar amb Interrupcions.

Altres prestacions, que no s'han utilitzat però se'n disposa, serien:

- mòdul intern convertidor AD de 10 bits amb 12 canals.
- mòdul de generació de senyal PWM.
- ICSP (*In-Circuit Serial Programming*), que permet l'escriptura del programa dins del IC sense extreure'l del circuit.
- 2 mòduls comparadors.
- 2 mòduls temporitzadors de 8/16 bits.

Veiem el diagrama i les funcions de cada pin, segons dades del fabricant:

PIC16F690 Pin Diagram (PDIP, SOIC, SSOP)

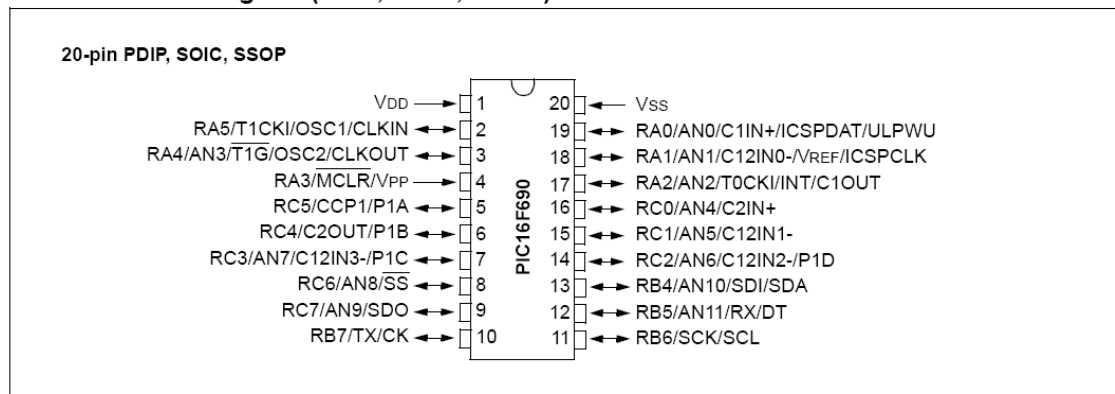


TABLE 5: PIC16F690 PIN SUMMARY

I/O	Pin	Analog	Comparators	Timers	ECCP	EUSART	SSP	Interrupt	Pull-up	Basic
RA0	19	AN0/ULPWU	C1IN+	—	—	—	—	IOC	Y	ICSPDAT
RA1	18	AN1/VREF	C12IN0-	—	—	—	—	IOC	Y	ICSPCLK
RA2	17	AN2	C1OUT	T0CKI	—	—	—	IOC/INT	Y	—
RA3	4	—	—	—	—	—	—	IOC	Y ⁽¹⁾	MCLR/VPP
RA4	3	AN3	—	T1G	—	—	—	IOC	Y	OSC2/CLKOUT
RA5	2	—	—	T1CKI	—	—	—	IOC	Y	OSC1/CLKIN
RB4	13	AN10	—	—	—	—	SDI/SDA	IOC	Y	—
RB5	12	AN11	—	—	—	RX/DT	—	IOC	Y	—
RB6	11	—	—	—	—	—	SCL/SCK	IOC	Y	—
RB7	10	—	—	—	—	TX/CK	—	IOC	Y	—
RC0	16	AN4	C2IN+	—	—	—	—	—	—	—
RC1	15	AN5	C12IN1-	—	—	—	—	—	—	—
RC2	14	AN6	C12IN2-	—	P1D	—	—	—	—	—
RC3	7	AN7	C12IN3-	—	P1C	—	—	—	—	—
RC4	6	—	C2OUT	—	P1B	—	—	—	—	—
RC5	5	—	—	—	CCP1/P1A	—	—	—	—	—
RC6	8	AN8	—	—	—	—	SS	—	—	—
RC7	9	AN9	—	—	—	—	SDO	—	—	—
—	1	—	—	—	—	—	—	—	—	VDD
—	20	—	—	—	—	—	—	—	—	VSS

Note 1: Pull-up activated only with external MCLR configuration.

Figura 7-19. Descripció dels pins del PIC 16F690 segons dades del fabricant.

Mòduls interns utilitzats: EUSART

El fet d'incorporar mòduls interns amb diverses funcions, ens permeten no utilitzar recursos de programari per aquestes tasques. És a dir, estan implementades internament a nivell de maquinari, podent així alliberar recursos de programa.

Un dels casos i de vital importància en el nostre projecte és el mòdul EUSART (*Enhanced Universal Synchronous Asynchronous Receiver Transmitter*). Gràcies a aquesta funcionalitat, podem connectar qualsevol dispositiu que disposi de connectivitat RS-232 amb el nostre microcontrolador. Aquest és el cas del nostre ordinador personal, però encara millor, a través del nostre dispositiu transceptor de ràdio.

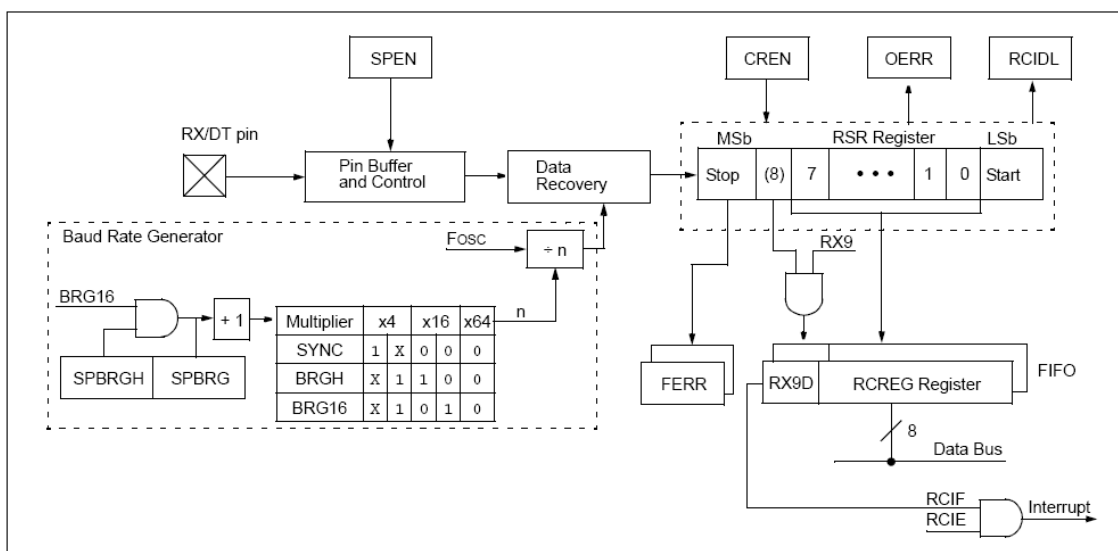


Figura 7-21. Diagrama de blocs del mòdul intern del receptor EUSART.

Aquest maquinari intern, ens permet accedir-hi a través del pin RX/DT que concretament es troba en el pin RB5 (12). Addicionalment, pot generar una interrupció que pot ser detectada i utilitzada a través del programari del microcontrolador.

El control d'aquest mòdul es realitza mitjançant uns registres específics, que tant es poden manipular en baix nivell (llenguatge ensamblador) com en alt nivell a través del compilador utilitzat i el llenguatge C.

Per al seu control existeixen els següents registres mostrats en la taula 3 registres:

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
BAUDCTL	ABDOVF	RCIDL	—	SCKP	BRG16	—	WUE	ABDEN	01-0 0-00	01-0 0-00
INTCON	GIE	PEIE	T0IE	INTE	RABIE	T0IF	INTF	RABIF	0000 000x	0000 000x
PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	-000 0000	-000 0000
PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
RCREG	EUSART Receive Data Register								0000 0000	0000 0000
RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
SPBRG	BRG7	BRG6	BRG5	BRG4	BRG3	BRG2	BRG1	BRG0	0000 0000	0000 0000
SPBRGH	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	BRG9	BRG8	0000 0000	0000 0000
TRISB	TRISB7	TRISB6	TRISB5	TRISB4					1111 ----	1111 ----
TXREG	EUSART Transmit Data Register								0000 0000	0000 0000
TXSTA	CSRC	TX9	TXEN	SYNC	SENDER	BRGH	TRMT	TX9D	0000 0010	0000 0010

Legend: x = unknown, — = unimplemented read as '0'. Shaded cells are not used for Asynchronous Transmission.

Taula 7-3. Registres implicats en el mòdul EUSART.

En el nostre cas, el compilador ens ofereix opcions per al control sense manipular directament els registres, i sense haver d'escriure codi en llenguatge ensamblador:

```
#use rs232(ERRORS,uart1,baud=19200,parity=N,xmit=PIN_B7,rcv=PIN_B5,bits=8)
```

Amb aquesta directiva del compilador, s'estableixen directament la velocitat de transmissió en 19200 bps, no paritat, els pins d'entrada i sortida, etc.

Aquest serà doncs la porta d'entrada de les dades provinents via ràdio des de l'exterior, concretament des de l'estació de terra. Veurem el seu funcionament detallat en el següent capítol 8.

Mòduls interns utilitzats: SSP (Synchronous Serial Port) o I²C™

El bus de comunicacions I²C™ (*Inter-Integrated Circuit*), és un bus sèrie síncron utilitzat majoritàriament en la comunicació entre diferents elements dins d'un mateix circuit. És el cas de sensors, memòries, displays, etc.

En el nostre projecte, té especial interès perquè serà la forma amb que hi connectarem i ens comunicarem amb el nostre sensor de posicionament, l'accel·leròmetre.

Aquest bus utilitza un protocol concret, havent definit primerament un dispositiu màster (el microcontrolador en el nostre cas) i un element esclau (el sensor de posició).

Afortunadament, el microcontrolador utilitzat, incorpora també físicament aquest mòdul, de nou per alliberar recursos de programa, tot i que també es podria implementar a través de funcions de programari en cas necessari o de no disposar-ne.

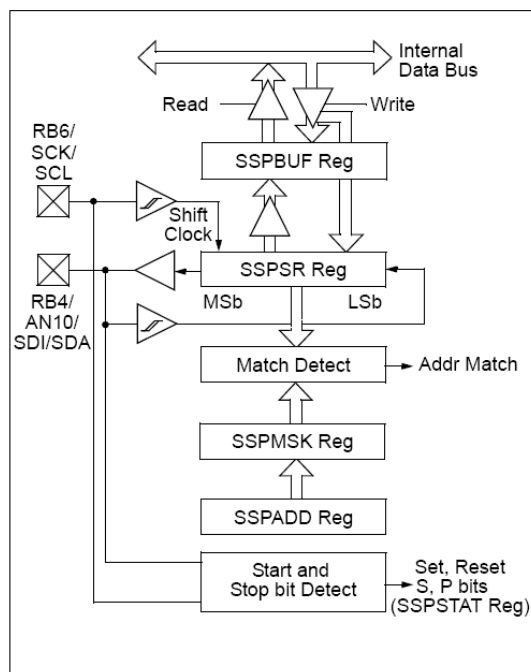


Figura 7-22. Diagrama de blocs del mòdul I²C intern.

En la figura 7-22 trobem el diagrama de blocs d'aquest maquinari implementat físicament. Com a punt fonamental, destaquem els pins d'entrada i sortida, que es corresponen amb RB6/SCL (13) i RB4/SDA (11).

De nou es disposa de tota una sèrie de registres, que cal utilitzar per a la configuració i utilització d'aquest mòdul. Es pot veure en la següent taula 7-4.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on POR, BOR	Value on all other Resets
0Bh/8Bh/10Bh/18Bh	INTCON	GIE	PEIE	TOIE	INTE	RABIE	TOIF	INTF	RABIF	0000 000x	0000 000x
0Ch	PIR1	—	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	-000 0000	-000 0000
13h	SSPBUF	Synchronous Serial Port Receive Buffer/Transmit Register								xxxx xxxx	uuuu uuuu
14h	SSPCON	WCOL	SSPOV	SSPEN	CKP	SSPM3	SSPM2	SSPM1	SSPM0	0000 0000	0000 0000
86h	TRISB	TRISB7	TRISB6	TRISB5	TRISB4	—	—	—	—	1111 ----	1111 ----
93h	SSPMSK ⁽²⁾	MSK7	MSK6	MSK5	MSK4	MSK3	MSK2	MSK1	MSK0	1111 1111	1111 1111
94h	SSPSTAT	SMP ⁽³⁾	CKE ⁽³⁾	D/Ā	P	S	R/W	UA	BF	0000 0000	0000 0000
8Ch	PIE1	—	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IF	TMR1IF	-000 0000	-000 0000

Legend: — = Unimplemented locations, read as '0', u = unchanged, x = unknown. Shaded cells are not used by the SSP module.

Taula 7-4. Registres implicats en el mòdul intern SSP I²C.

A l'igual que amb el mòdul EUSART, el nostre compilador ens permet accedir a aquests registres, a través de directives concretes.

```
#use I2C (MASTER, SCL=PIN_B6, SDA=PIN_B4, SLOW)
```

Així, per exemple, en el nostre codi, establim el microcontrolador com a màster, definim els pins d'entrada i sortida i utilitzem una velocitat en el bus concreta.

Per tant, ja sabem també com ens connectarem amb el nostre sensor de moviment. Explicarem el detall també en el capítol 8.

7.3.4. Sensor de Posició (Acceleròmetre)

Un acceleròmetre, com el seu nom indica, mesura acceleració. És a dir, com varia la velocitat d'un cos. N'hi ha de diferents tipus, però avui en dia, gràcies a la microelectrònica, poden gaudir dels MEMS (*Microelectromechanical Systems*).

Es tracta de micro-màquines fabricades utilitzant la tecnologia ja avançada de la microelectrònica, de manera que es pot integrar dins d'un únic circuit integrat un element mecànic de silici que fa de transductor mecànic, conjuntament amb l'electrònica necessària per tal d'acondicionar i comunicar els valors mesurats convertits en senyals, cap a l'exterior.

L'element triat en el nostre cas, és un sensor anomenat digital, pel fet que la seva sortida és ja transformada en senyal digital. D'aquesta forma, el podem connectar directament a microcontroladors o altres dispositius sense la necessitat d'electrònica addicional per adaptar senyals analògics.

Es tracta del model **MMA7455** de la firma **freescalsemiconductor**, un acceleròmetre digital de tres eixos que, afortunadament, s'ha aconseguit muntat ja en una placa en format DIP per a poder ser utilitzar en el nostre prototipus de disseny.

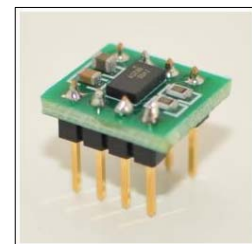


Figura 7-23. Acceleròmetre en format DIP utilitzat.

S'ha triat aquest model per reunir les característiques necessàries, així com per el seu cost assequible.

Les principals funcions que incorpora són

- Sortida digital SPI/ I²C,
- Possibilitat de seleccionar diferents rangs de mesura ($\pm 2g/\pm 4g/\pm 8g$)
- Tensió d'alimentació de 3.3v
- 8 bits de resolució
- Sensibilitat de 64 LSB/g
- Baix consum de corrent: 400 μA

La següent figura ens mostra la distribució de pins, tot i que no tots seran accessibles en el nostre cas degut al muntatge fet en la placa DIP. No obstant, accedirem perfectament als necessaris, que són SDA / SCL, alimentació i GND.

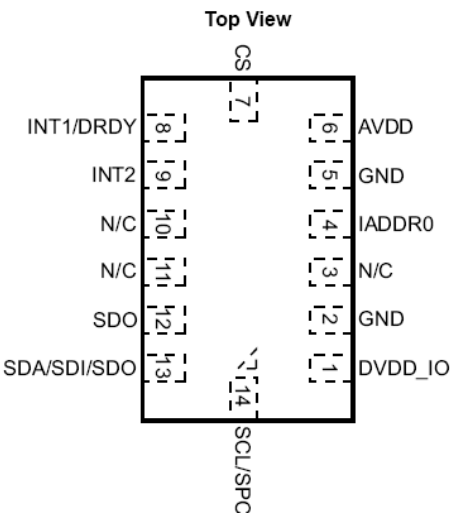


Figure 12. Pinout Description

Pin #	Pin Name	Description	Pin Status
1	DVDD_IO	Digital Power for I/O pads	Input
2	GND	Ground	Input
3	N/C	No internal connection. Leave unconnected or connect to Ground.	Input
4	IADDR0	I ² C Address Bit 0	Input
5	GND	Ground	Input
6	AVDD	Analog Power	Input
7	CS	SPI Enable (0), I ² C Enable (1)	Input
8	INT1/DRDY	Interrupt 1/ Data Ready	Output
9	INT2	Interrupt 2	Output
10	N/C	No internal connection. Leave unconnected or connect to Ground.	Input
11	N/C	No internal connection. Leave unconnected or connect to Ground.	Input
12	SDO	SPI Serial Data Output	Output
13	SDA/SDI/SDO	I ² C Serial Data (SDA), SPI Serial Data Input (SDI), 3-wire interface Serial Data Output (SDO)	Open Drain/ Input/ Output
14	SCL/SPC	I ² C Serial Clock (SCL), SPI Serial Clock (SPC)	Input

Figura 7-24. Descripció de pins del component utilitzat.

La lectura que ens dóna, a través del bus I²C, correspon a un valor de 8 bits, en funció de l'acceleració, i en el nostre cas en funció de la inclinació en els seus eixos, que serà el que haurem de mesurar.

Disposa de diferents modes de funcionament que s'hauran de configurar abans de treballar amb ell, i es farà utilitzant el mateix bus de comunicacions. En concret, nosaltres haurem de seleccionar **mode mesurament**, i triarem un rang de mesura de **2g**, la qual implica una sensibilitat de **64 LSB/g**.

Table 6. Configuring the Mode using Register \$16 with MODE[1:0] bits.

MODE [1:0]	Function
00	Standby Mode
01	Measurement Mode
10	Level Detection Mode
11	Pulse Detection Mode

\$16: Mode Control Register (Read/Write)

D7	D6	D5	D4	D3	D2	D1	D0	Bit
--	DRPD	SPI3W	STON	GLVL[1]	GLVL[0]	MODE[1]	MODE[0]	Function
0	0	0	0	0	0	0	0	Default

Table 5. Configuring the g-Select for 8-bit output using Register \$16 with GLVL[1:0] bits.

GLVL [1:0]	g-Range	Sensitivity
00	8g	16 LSB/g
01	2g	64 LSB/g
10	4g	32 LSB/g

Figura 7-25. Modes de funcionament a seleccionar.

Quant al connexionat, serà força simple i només haurem de tenir en compte utilitzar les resistències de pull-up que faran falta en el bus de comunicació. El fabricant ens mostra un exemple de connexió, que serà el quin utilitzarem.

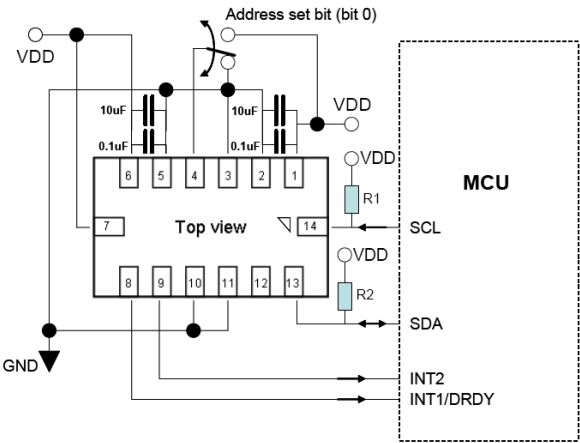


Figura 7-26. Exemple de connexionat donat pel fabricant.

Pel que fa al bus de comunicació, caldrà seguir estrictament les indicacions donades pel fabricant:

- Adreça \$1D.
- Funcionant en mode esclau.
- Protocol, que només utilitzarem en els formats que es mostren en les figures 7-27 i 7-28.

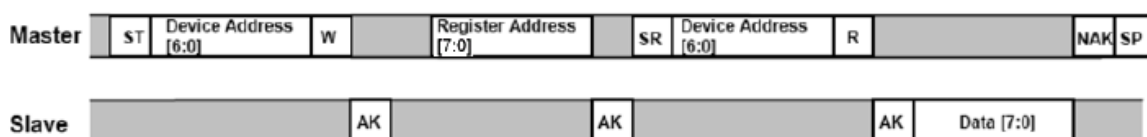


Figura 7-27. Protocol de lectura de múltiples bytes.

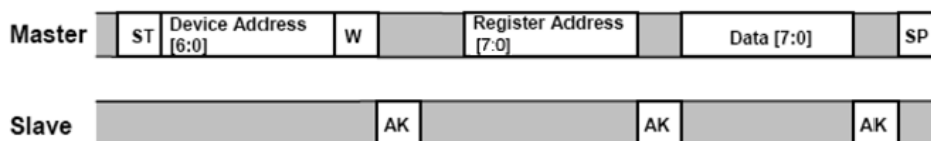


Figura 7-28. Protocol d'escriptura de byte únic.

Posteriorment, en el capítol 8 veurem com es realitza la inicialització i com es fan les diferents lectures a través del programari del microcontrolador.

7.3.5. Esquema i Muntatge del maquinari al vehicle

El fet d'utilitzar una placa d'avaluació per al cas de la FPGA limita en part la realització d'un disseny compacte i reduït. No obstant això s'ha volgut valorar el fet de poder treballar amb una eina de desenvolupament tant poderosa i deixant per a posteriors projectes la implementació en una única PCB per al conjunt complet.

D'aquesta forma, s'ha dissenyat només maquinari per a la part complementària del conjunt, és a dir, el microcontrolador, el transceptor (receptor) de ràdio i l'acceleròmetre.

En les figures 7-29 i 7-30 es pot veure el disseny de l'esquema d'aquest circuit, així com un el disseny realitzat per al circuit imprès.

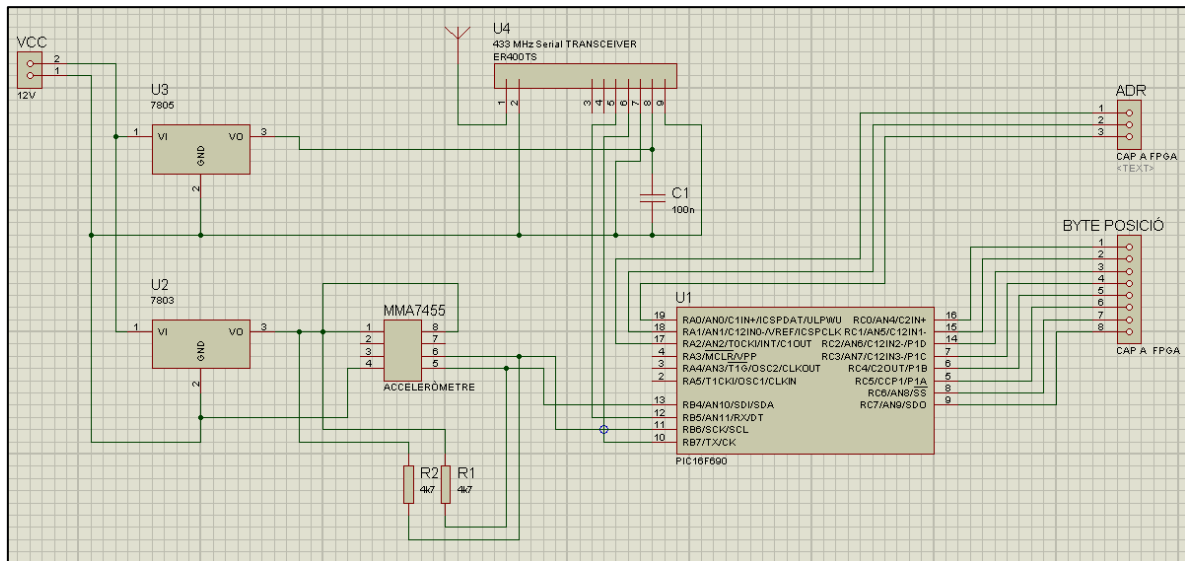


Figura 7-29. Esquema del muntatge complementari a la FPGA.

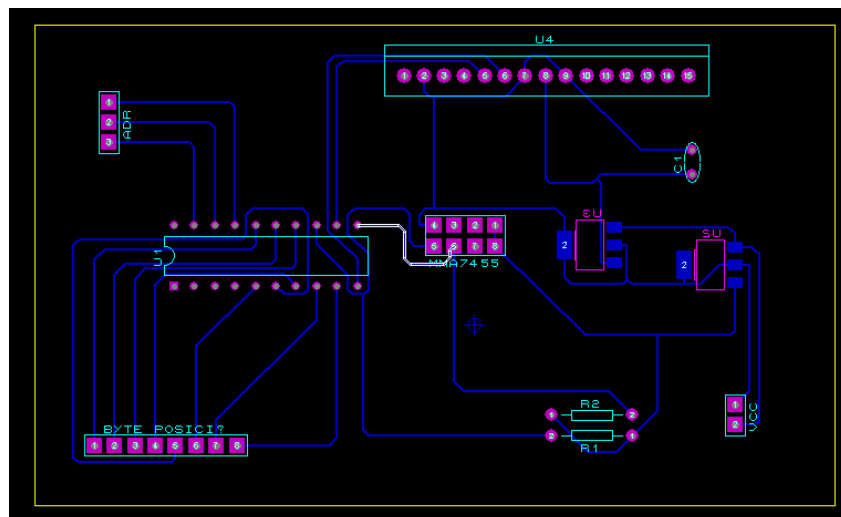


Figura 7-30. Disseny del circuit imprès.

7.3.6. Programari per als diferents components del Vehicle

Per al disseny dels diferents elements de programari s'han utilitzat dos tipus d'entorn de programació.

Pel que fa al microcontrolador el llenguatge triat ha estat el C, pel fet d'haver estat utilitzat durant la carrera i per la comoditat envers del llenguatge ensamblador. No obstant, reconeixent les limitacions que es donen al no tenir un estricte control i coneixement dels registres i recursos que es manipulen amb un llenguatge d'alt nivell en aquest cas, envers del llenguatge ensamblador.

L'entorn utilitzat per a desenvolupar el disseny d'aquest codi, ha estat el compilador CCS, juntament amb el simulador PIC C Compiler.

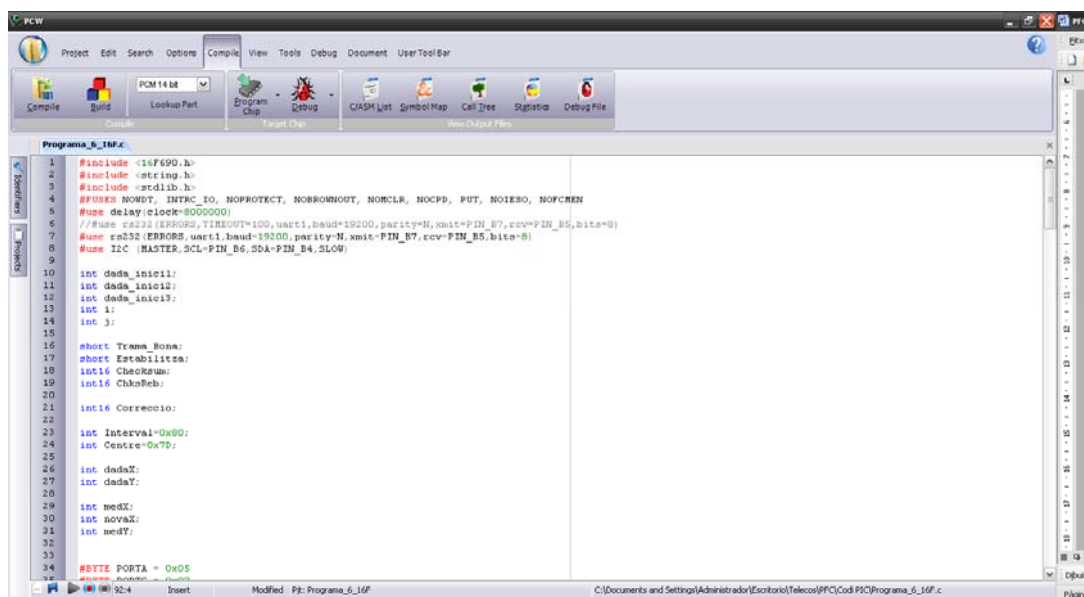


Figura 7-31. Imatge principal de l'entorn de programació CCS.

Cal tenir en compte, que a l'igual que altres usuaris, hem pogut constatar que aquest entorn conté alguns errors que poden dificultar el disseny, provocant, en alguns casos resultats inesperats. Ens consta també que s'estan solucionant en versions actualitzades.

Un cop verificat i compilat, el programari per al microcontrolador ha estat bolcat dins del pic mitjançant el programa gratuït de Microchip **PICKit 2**.

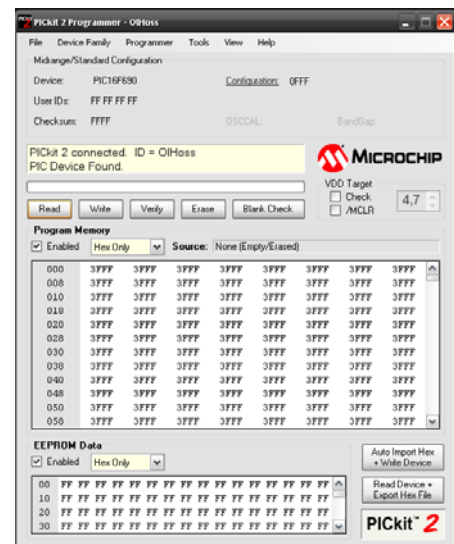


Figura 7-32. Programari de Microchip, utilitzat en l'escriptura del microcontrolador.

Pel que fa a la FPGA, s'ha utilitzat el programari que posa a disposició el propi fabricant per al desenvolupament, anomenat *ispLever Project Navigator*.

Es tracta d'un entorn força complet, ahora que complex, però que soluciona des de l'escriptura del codi, passant per la simulació, l'assignació dels pins d'entrada i sortida i la final escriptura de l'arxiu .jed en el component.

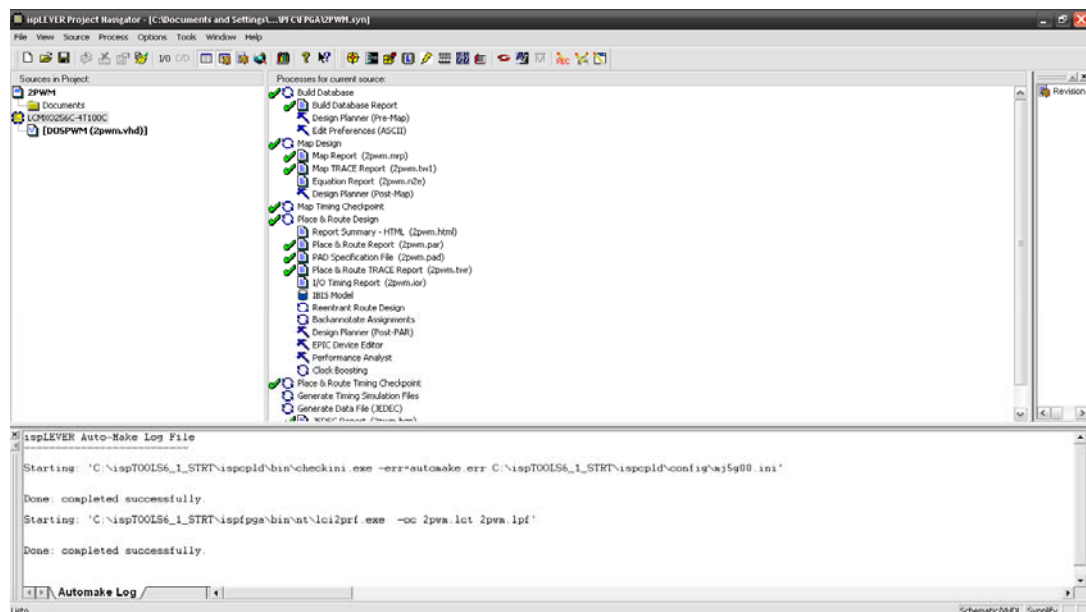


Figura 7-33. Imatge principal de l'entorn de programació ISP Lever Project Navigator.

8. DESENVOLUPAMENT I FUNCIONAMENT

Després d'haver donat la idea general i haver presentat tots els component que componen el sistema, ja es poden anar enllaçant a través de la seva interconnexió i funcionalitats.

Partirem des de l'estació de terra i s'aniran seguint els senyals explicant com es generen i com interactuen entre ells.

8.1. PROTOCOL DE COMUNICACIONS

En primer lloc, cal tornar a l'aplicació de control per agafar la informació del joystick i tornar a veure la trama de comunicacions generada.

S'havia explicat a 7.1.2 com el programari llegia del joystick una informació de 16 bits. És a dir, valors entre 0 i 65.535 per a cada un dels 4 eixos dels quals disposem.

La informació representada en aquests 16 bits s'ha de convertir a 8 bits. Això s'explica per dos motius.

En primer lloc des del punt de vista del pilotatge, tot el recorregut de la palanca s'ha d'assignar a tot el possible recorregut d'un servo en el cas extrem. Com es va explicar a 7.3.1 el màxim recorregut possible és de 180°. D'aquesta forma, amb una resolució de 8 bits (255 valors) obtenim

$$\frac{180^\circ}{255} = 0,71^\circ$$

És a dir, el mínim moviment que es pot executar en un servo a través del joystick amb representació de 8 bits és de 0,71°, valor que és més que suficient per al pilotatge.

Tot i així, es veurà a continuació (en l'apartat 8.2) que degut al interval del senyal PWM reconegut per un servo, el rang màxim de valors quedarà reduït a 205, per tant

$$\frac{180^\circ}{205} = 0,88^\circ$$

Que encara és prou vàlid per a uns moviments prou afinats. De fet, encara més, gairebé mai es treballa amb els 180° , amb la qual cosa tornem a millorar i molt la resolució obtinguda amb 8 bits.

En segon lloc, quan generem el senyal PWM amb el circuit explicat a 8.2 es veurà que es necessita un registre de posició de 8 bits.

Per tant, en primer lloc, es divideix el valor capturat del joystick entre 2^8 de forma que se n'obté un valor llegit de 8 bits, entre 0 i 255 o 0 i FFh

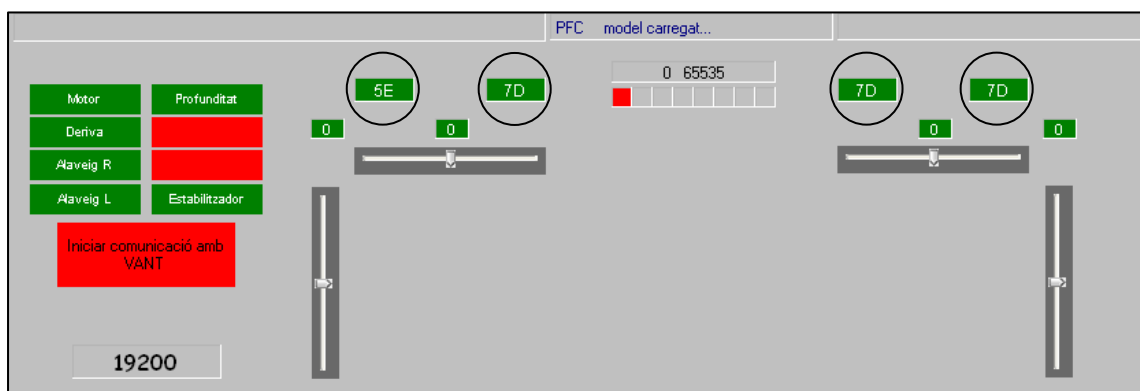


Figura 8-1. Lectures del joystick, per al 4 primers canals, en valors hexadecimal.

Un cop tenim ja aquests valors, el següent pas és adaptar-los a la configuració. Com que tal i com s'havia dit, aquest programari no està dissenyat per al projecte, ens limitem a explicar que, aquests valors llegits per a cada canal, són adaptats als recorreguts reals i mecànics del servo. Però tot i això seguiran sent un valor de 8 bits que d'alguna forma caldrà fer arribar fins al seu corresponent actuador.

Sí que expliquem com s'ha adaptat i generat la trama de dades necessària per a una transmissió correcta i fiable.

Recuperem per això la taula exposada al capítol 7.1.3. on s'hi representa la trama generada a l'estació de terra.

Hi són clarament diferenciables els valors del bytes 4..11, que corresponen a les posicions obtingudes a partir del programa de control i el joystick. En el cas de que algun canal no s'utilitzi, el programa hi assigna el valor 0.

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10	Byte 11	Byte 12	Byte 13
FF	FE	FD	Pos Canal 1	Pos Canal 2	Pos Canal 3	Pos Canal 4	Pos Canal 5	Pos Canal 6	Pos Canal 7	Pos Canal 8	CHK 1	CHK 2
Sincronització de trama			Dades de posició de cada canal								Checksum calculat	

Com que aquesta trama necessita ser enviada per ràdio, i la comunicació és asíncrona, calia algun tipus de sincronització per tal que cada canal s'assigni correctament al seu actuator. Per aquest motiu es va haver d'afegir els tres primers bytes. Corresponen únicament a tres valor triats per tal que el nostre receptor els reconegui com a inici de trama. Només les trames que continguin aquests tres valors inicials seran considerades com a vàlides abans d'anar als seus servos corresponents.

Taula 8-1. Trama de comunicacions enviada des de la base a l'aeronau.

Tot i aquesta contramesura, algunes de les trames arriben als actuadors corruptes tot i contenir els tres primers bytes d'inici de trama. Això genera moviments aleatoris i indesitjats en els servos que poden fer difícil o impossible el pilotatge.

Per aquest motiu, s'afegeix una filtre extra de verificació. Aquesta nova mesura consisteix en generar un checksum del valor de tots els bytes de la trama generada a terra. Aquest valor es calcula a través del programa de control i és afegit a la trama en els bytes 12 i 13 per a ser també enviats. D'aquesta forma, el receptor (el microcontrolador en el nostres cas) farà la mateix càlcul i comparà el valor de suma calculat amb el valor de suma rebut.

Només si són idèntics, es donarà per bona la trama abans d'enviar-la als servos per a produir el moviment.

S'ha vist per tant com està implementada una trama completa de comunicacions. Aquesta és enviada en intervals teòrics de 50 ms, a través del sistema de ràdio que ja hem vist al capítol 7.2.

No obstant, de cara al transmissor (ordinador personal) i de cara al receptor (microcontrolador), aquesta informació rebuda, correcta o incorrectament és tractada com a dades enviades a través de protocol RS-232. És a dir, que per a aquests dos elements el sistema es comporta com si estiguessin connectats només a través d'un cable RS232 de tipus DB9.

8.2. PROTOCOL RS-232 (*Recommended Standard 232*)

No s'explicarà en detall, però donarem alguna noció del seu funcionament. Afortunadament per a nosaltres, es tracta d'un protocol tant utilitzat que molts dispositius electrònics l'incorporen per al seu ús. És el cas del nostre microcontrolador, cosa que facilita molt el disseny.

Abans de l'USB era un dels protocols més utilitzats per a la comunicació de dades en sèrie, de forma asíncrona. Aquest fet de la no sincronització entre el receptor RX i el transmissor TX, simplifica el disseny pel fet de no necessitar un senyal de rellotge comú entre els dos elements, que es poden trobar a distàncies concretes. Per altra banda cal definir un protocol per tal que la comunicació es faci correctament.

El protocol, força simple, es mostra a la següent figura 8-2.

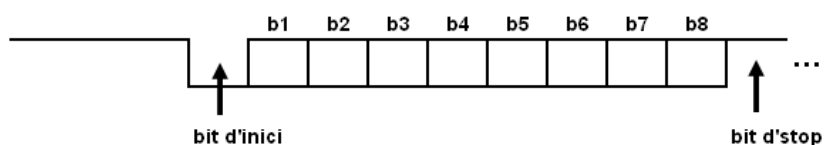


Figura 8-2. Protocol de comunicacions sèrie RS232.

Tot i que es pot arribar a complicar més, perquè existeixen definits altres tipus de senyals per al control de la comunicació entre perifèrics més complexes, en essència es pot fer la comunicació amb un únic cable connectat entre el transmissor TX i el receptor RX.

Cal per això definir uns paràmetres molt estàndards que són la velocitat de transmissió, el tipus de paritat, el número de bits de dades i el número de bits d'stop.

En el nostre cas s'ha utilitzat:

Baud rate: 19200

Paritat: NONE (Cap paritat utilitzada)

Bits de dades: 8

Bits d'stop: 1

Configuració VisualBasic.net utilitzada al transmissor (ordinador personal):

```
SerialPort1.BaudRate = 19200
SerialPort1.Parity = IO.Ports.Parity.None
SerialPort1.DataBits = 8
SerialPort1.StopBits = IO.Ports.StopBits.One
```

Configuració C utilitzada al receptor (microcontrolador):

```
#use rs232(ERRORS,uart1,baud=19200,parity=N,xmit=PIN_B7,rcv=PIN_B5,bits=8)
```

Aquesta és la configuració utilitzada tant per part del transmissor com del receptor.

8.3. RECEPCIÓ DE TRAMES AL MICROCONTROLADOR

Un cop les dades ja es troben en el port sèrie del microcontrolador, aquestes s'han de llegir i validar abans de ser utilitzades.

Per fer-ho hem utilitzat la següent rutina d'interrupció, que només s'activa quan arriben dades noves pel port sèrie del microcontrolador. D'aquesta forma el PIC disposa de temps extra per a executar altres tasques dins del programa.

```
#int_RDA
RDA_isr(){
    Checksum=0;
    Trama_Bona=0;

    dada_inici1=getc();           // Sincronitzem amb inici de trama (255) & (254) & (253)
    if (dada_inici1==0xFF)
    {
        dada_inici2=getc();
        if (dada_inici2==0xFE)
        {
            dada_inici3=getc();
            if (dada_inici3==0xFD)
            {
                for (i=0;i<=9;i++)
                    trama[i]=getc();

                for(i=0;i<=7;++i){
                    Checksum=Checksum+trama[i];    // Calculem el Checksum de la trama rebuda
                }

                ChksReb=(long)trama[8]*100+trama[9];    // Obtenim el checksum rebut del PC

                if((Checksum==ChksReb)&(Checksum!=0))
                {
                    //printf(".");
                    return Trama_Bona=1;
                }
                else
                {
                    //printf("*");
                    return Trama_Bona=0;
                }
            }
        }
    }

    return Trama_Bona=0;
}
```

Només s'executarà aquesta rutina d'interrupció en el cas de rebre dades noves. La funció d'aquest codi serà, per una banda obtenir caràcter a caràcter tots els bytes de la trama, i per l'altra, validar i assegurar que es tracta de la mateixa trama que ha estat enviada, mitjançant el càlcul del checksum.

8.3.1. Obtenció de la trama de comunicació

Mitjançant els bucles *if-then* s'esperen i s'emmagatzemen els caràcters rebuts fins a completar la seqüència 'FF', 'FE', 'FD'. Només si es dóna aquesta circumstància, s'agafaran els 10 bytes següents i es guardaran dins la matriu *trama[]* per a la seva validació. Arribats aquí, la trama guardada no és encara vàlida fins a verificar el seu checksum.

En el cas que la seqüència inicial no sigui correcta, es sortirà del bucle d'interrupció omplint la variable *Trama_bona* amb el valor '0', indicant que la trama no és vàlida i no es pot fer servir.

8.3.2. Validació de la trama amb el checksum

Quan la seqüència de sincronització ha estat correcta, s'han guardat els següents 10 bytes dins la matriu de bytes *trama[]*. No obstant no hi ha garanties de que la trama enregistrada en la aquesta matriu sigui la mateixa que s'ha enviat des de l'estació de terra. En molts casos la trama és corrupta tot i haver complert correctament la seqüència de sincronització.

Per tal de validar això, es calcula la suma dels bytes 1 fins al 8 de la matriu de dades *trama[]*. Aquest valor obtingut de la suma es compara amb el rebut, que es troba contingut en *trama[8]* i *trama[9]* (l'índex de la matriu comença pel valor 0). Per fer-ho cal fer la operació

$$\text{Checksum rebut} = (\text{trama}[8] \cdot 100) + \text{trama}[9]$$

Aquesta operació és necessària perquè la suma de 8 valors de 8 bits necessita 2 bytes per a la seva representació. Aquests 2 bytes han de ser separats per enviar-se, i tornats a calcular en la recepció. Això és el que fa aquesta operació, inversament com s'ha fet en el càlcul fet pel programa de control abans d'enviar-se.

Un cop superat aquest filtre, i conseqüentment l'anterior, el valor de la variable *Trama_bona* es posa a '1' indicant a la resta del programa que la trama és bona per al seu ús.

8.4. PROTOCOL MICROCONTROLADOR – FPGA

Abans de veure què fa el programa del microcontrolador quan té emmagatzemada una trama correcta, convé explicar el procediment de maquinari mitjançant el qual el microcontrolador es comunica amb la FPGA. Avancem així aleshores, que l'encarregada de generar els diferents senyals PWM a partir dels valors emmagatzemats dins la variable *trama[]* del microcontrolador serà precisament la FPGA.

Si ampliem el diagrama de blocs vist a la figura 5-1, obtenim el següent muntatge:

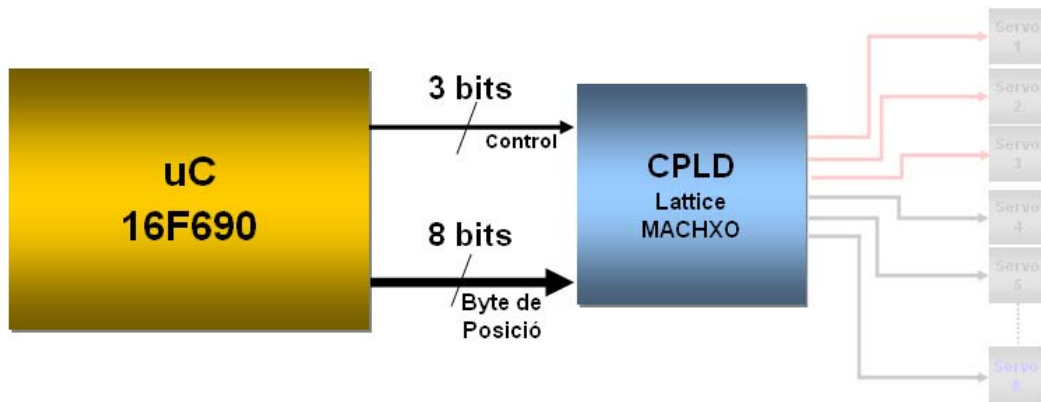


Figura 8-3. Protocol de comunicació microcontrolador – FPGA.

Per tal que la informació de posició enviada des del pilot arribi fins al corresponent actuador, el microcontrolador es comunica amb la FPGA qui haurà de generar el senyal necessari però només per l'actuador necessari. Per fer-ho, el PIC no només li diu a la FPGA quin és el valor d'aquest byte de posició, sinó també a quin servo va dirigit aquest valor.

Aquest procés s'ha implementat mitjançant dos senyals entre aquests dos elements.

D'una banda, totes les 8 sortides disponibles al port C del microcontrolador queden assignades per enviar en paral·lel aquest valor a la FPGA.

D'altra banda tres línies addicionals del port A enviaran a la FPGA quin és el servo en concret que espera el byte de posició enviat pel port C.

Aquest valor no és més que l'índex i que ocupa el valor del byte de posició dins de la matriu trama[i]. Posteriorment, al capítol 8.6 es veurà com la FPGA utilitza aquesta informació.

8.4.1. Enviament dels valors a la FPGA

Per tant, per enviar aquesta informació des del microcontrolador cap a la FPGA, només cal situar-la en els corresponents ports de sortida. Un bucle *for*, que només s'executa quan la trama és vàlida, va col·locant un a un els diferents valors per al registre de posició, acompanyats de l'índex que indica a quin actuator van adreçats.

```
// ***** ENVIEM DADES A LA FPGA *****  
for(j=0;j<=7;++j)  
{  
    PORTA = j;  
    PORTC = trama[j];  
}
```

D'aquesta forma, la FPGA ja té en les seves corresponents entrades la informació necessària. Els següent pas, serà explicar com la FPGA utilitza aquests valors per a realitzar la seva funció, que és generar el senyal PWM corresponent a cada una de les seves sortides, independentment.

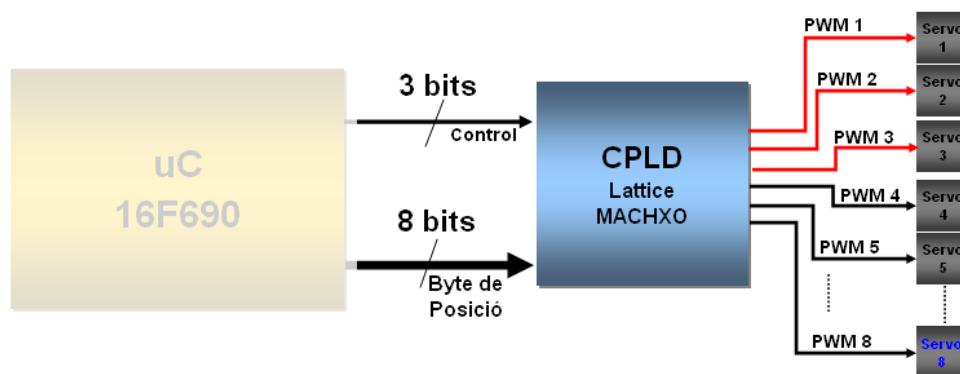


Figura 8-4. Diagrama de blocs per la funció de la FPGA.

8.5. GENERACIÓ del SENYAL PWM

Un dels mètodes possibles per tal de generar el senyal PWM és utilitzar els propis recursos del microcontrolador. Per fer-ho, cal utilitzar els temporitzadors adaptant correctament la freqüència de l'oscil·lador intern, així com els paràmetres necessaris. Caldria assignar una sortida per a cada dispositiu extern, és a dir, un port sencer de 8 sortides.

A priori potser és una tasca senzilla, però aviat es complica si el que volem és generar 8 senyals diferents i independents entre elles. No és que no es pugui, ni molt menys, però per diversos motius es va optar per utilitzar maquinari extra.

Aquesta opció permet alliberar recursos del microcontrolador i dedicar-se millor a l'execució del programa.

Una de les opcions utilitzades correspon al següent circuit digital mostrat a la figura 8-5.

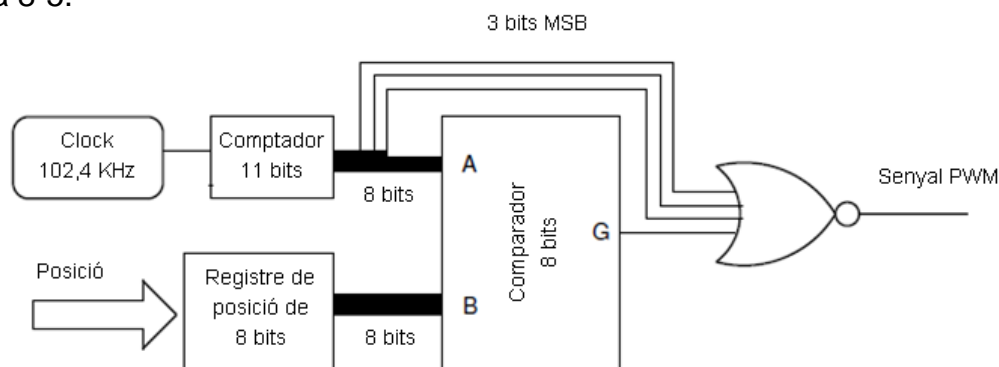


Figura 8-5. Circuit digital per a la generació del senyal PWM.

Expliquem el seu funcionament a continuació.

Per una banda tenim un comptador d' 11 bits, que funcionant amb un senyal de rellotge de 102,4 KHz és capaç de generar un pols de període 20ms.

$$\frac{1}{102,4 KHz} = 9,766 \cdot 10^{-6} s \quad \text{que multiplicat per } 2^{11}$$

s'obté el període del pols PWM = 0,02 s = 20 ms

D'una altra banda, tenim un registre de posició de 8 bits. Aquest registre correspondrà amb el que ens dona el microcontrolador (Byte de posició) i equival a un valor entre 0 i 255. De fet, es pot assimilar directament al valor que ens donaria el joystick treballant al 100% del seu recorregut.

És important el següent càlcul:

$$2^8 \cdot 9,766 \cdot 10^{-6} \text{ s} = 2,5 \cdot 10^{-3} \text{ s}$$

És a dir, dins dels diferents valors que es poden obtenir amb el registre de 8 bits, tindrem el màxim i el mínim interval del T_{on} del nostre senyal PWM.

Més concretament tindrem el màxim amb

$$\frac{2,3 \cdot 10^{-3} \text{ s}}{9,766 \cdot 10^{-6} \text{ s}} = 235,52 \rightarrow 236d = ECh$$

i el mínim amb

$$\frac{0,3 \cdot 10^{-3} \text{ s}}{9,766 \cdot 10^{-6} \text{ s}} = 30,72 \rightarrow 31d = 1Fh$$

Si continuem amb el circuit de la figura 8-5, el que hi tenim ara és un comparador de 8 bits, que compara el valor de 8 bits del registre de posició amb els 8 bits menys significatius (LSB) del comptador de 11 bits. A la sortida del comparador tindrem un '0' mentre el valor del comptador no superi el valor del registre de posició. En el moment en que s'igualin o es superi, la sortida del comparador passa a ser '1'.

Aquest '1' de la sortida del comparador juntament amb els 3 bits més significatius (MSB) del comptador d'11 bits, es troben en una porta NOR. Aquesta donarà a la seva sortida un '1' mentre totes les seves entrades siguin '0'. És a dir, abans que el comptador arribi al valor del registre de posició, o bé, abans que el comptador arribi al valor 256.

Després de qualsevol d'aquests dos casos, la sortida passa a ser '0'.

Tenim així generat el nostre senyal PWM amb el temps de període necessari. I segons els càlculs, el nostre registre de posició hauria d'estar entre els valors 1Fh i ECh, per a complir les especificacions del fabricant dels servos.

El següent pas és generar aquest senyal tantes vegades com es necessiti, havent decidit utilitzar 8 canals, és a dir, 8 senyals independents.

S'havia descartat també generar-ho a partir del microcontrolador. Per tant ens queda la opció d'implementar el circuit amb components discrets, un total de vuit circuits, o bé com és el cas, utilitzar dispositius de maquinari programable com una CPLD o una FPGA, mitjançant programació en VHDL.

8.6. Disseny del circuit en VHDL

Un cop s'ha decidit implementar tota aquesta circuiteria digital mitjançant la "descripció hardware" anem a explicar el disseny, la funció del qual anirà tota adreçada a la FPGA ó CPLD.

Primerament, donem una visió exterior, des del punt de vista d'allò que es necessita connectar dins el nostre sistema.



Figura 8-6. Estructura "entity" del Disseny implementat amb la FPGA.

La figura 8-6 ens dóna una idea de la visió externa del conjunt que es dissenyarà mitjançant la FPGA.

El senyal **Adr** es correspon amb el que hem anomenat anteriorment com a **control**, és a dir, el que ens indica a quin actuator està adreçat el byte de posició.

Per tant, ja es pot veure aquesta entitat en el codi VHDL:

```
entity DOSPWM is
port(BYTE_POS: in std_logic_vector ( 7 downto 0 );
      CLK: in std_logic;
      ADR: in std_logic_vector(2 downto 0);
      PWM0: out std_logic;
      PWM1: out std_logic;
      PWM2: out std_logic;
      PWM3: out std_logic;
      PWM4: out std_logic;
      PWM5: out std_logic;
      PWM6: out std_logic;
      PWM7: out std_logic);
end;
```

Seguidament es mostra en la figura 8-7, un esquema general de tot el circuit que posteriorment s'escriurà en VHDL. Ja s'havia vist al capítol 8-5 el circuit digital que genera cada senyal, per tant veiem la resta del circuit necessari:

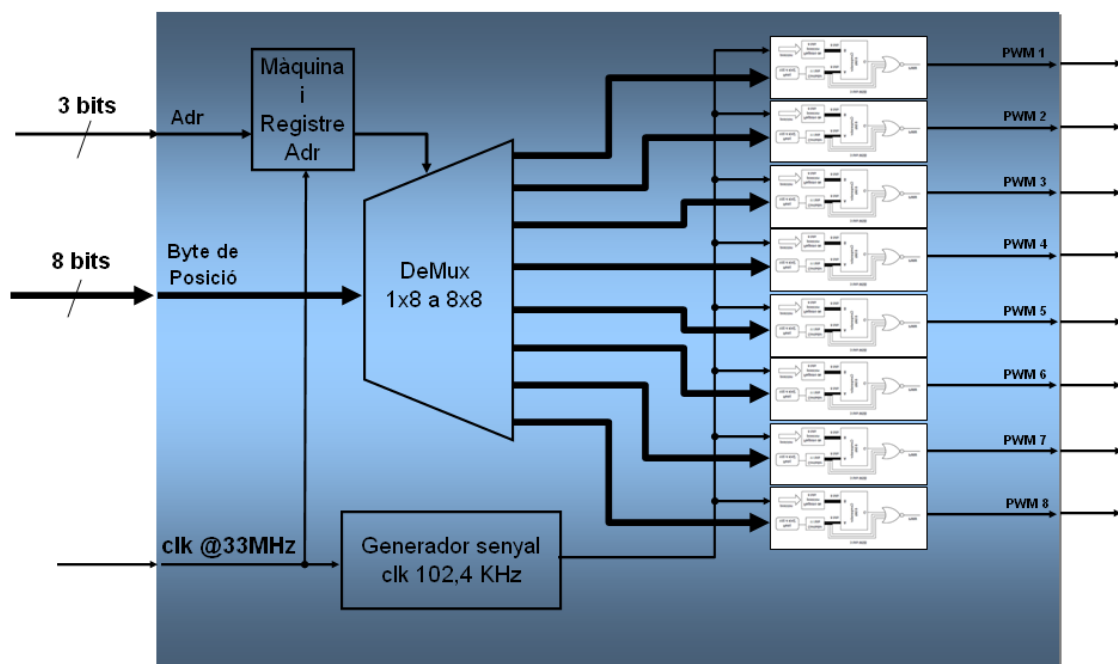


Figura 8-7. Circuit intern de la FPGA, desenvolupat en VHDL.

A continuació expliquem cada un dels blocs, que es convertiran en *processos* de codi VHDL.

Donat que el microcontrolador passa a la FPGA, tant el byte de posició com l'ordre dins la trama (ara anomenat **Adr**), cal utilitzar un circuit demultiplexor per tal d'adreçar l'entrada en aquell moment cap a la sortida de l'actuador corresponent. Aquesta adreça és precisament el senyal **Adr** que ens dona el microcontrolador.

8.6.1. Generador del senyal de clk de 102,4 KHz

Per a generar el senyal de rellotge necessari en els mòduls PWM, s'utilitza un divisor de freqüència realitzat amb un comptador.

$$\frac{33MHz}{102,4KHz} = 322,27$$

El senyal de rellotge d'entrada de la placa d'avaluació és de 33 MHz. Si el que necessitem és un senyal de 102,4 KHz caldrà que el comptador s'incrementi 323 vegades i canviï aleshores el seu estat. Per tal que aquest rellotge tingui un 50% de rendiment de cicle, farem el canvi de flanc en la meitat d'aquest compte és a dir, quan arribem als 162 comptatges.

Això és el que fa el següent procés VHDL que té com a entrada un senyal clk de 33 MHz i un de sortida clk2, de 102,4 KHz.

```
process(CLK)
begin
  if CLK 'event and CLK='1' then
    if compta=323 then
      compta<=0;
    else
      compta<= compta+1;
    end if;
  end if;
end process;
process(CLK)
begin
  if CLK 'event and CLK='0' then
    if compta=0 then
      clk2<='1';
    elsif compta=162 then
      clk2<='0';
    end if;
  end if;
end process;
```

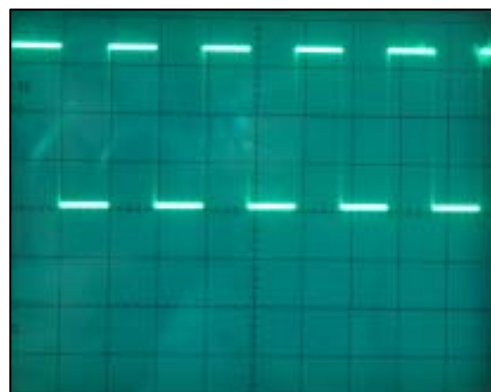


Figura 8-8. Senyal de rellotge de 102,4 KHz generat, vist a l'oscil·loscopi amb una base de temps de 10µs.

8.6.2. Màquina – Registre *Adr* i DeMultiplexor

Els dos senyals que ens envia el microcontrolador no arriben en el mateix instant a la FPGA. Per aquest motiu, no hi ha cap garantia que el byte de posició que hi ha a l'entrada del demultiplexor es correspongui amb el senyal ***Adr*** que hi hagi en aquest moment.

Per tal d'evitar aquest problema, es decideix enregistrar el senyal ***Adr*** per verificar que hagi canviat correctament, alhora que es dóna temps perquè s'estabilitzi l'entrada corresponent al byte de posició.

Aquest canvi correcte equival simplement a mirar que el nou valor rebut de ***Adr*** sigui una unitat superior al que hi ha enregistrat. D'aquesta forma, qualsevol altre valor queda descartat i la sortida del demultiplexor no és actualitzada.

Aquesta funció és el que realitza aquest procés.

```
process (ADR,ADR_Reg,BYTE_POS,clk)
begin
    if clk 'event and clk='0' then
        if ADR = Adr_Reg + 1 then
            Adr_Reg<=ADR;
            case Adr_Reg is
                when "000" => Byte_Pos_Reg0<= BYTE_POS;
                when "001" => Byte_Pos_Reg1<= BYTE_POS;
                when "010" => Byte_Pos_Reg2<= BYTE_POS;
                when "011" => Byte_Pos_Reg3<= BYTE_POS;
                when "100" => Byte_Pos_Reg4<= BYTE_POS;
                when "101" => Byte_Pos_Reg5<= BYTE_POS;
                when "110" => Byte_Pos_Reg6<= BYTE_POS;
                when "111" => Byte_Pos_Reg7<= BYTE_POS;
            end case;
        end if;
    end if;
end process;
```

8.6.3. Procés modulador del senyal PWM

Per a finalitzar amb el codi VHDL, dels components mostrats en la figura 8-7, ens queda només descriure els mòduls que generen el senyal PWM.

Aquest mòdul, que alhora es troba subdividit en diferents processos, està implementat 8 vegades dins de la circuiteria de la FPGA. Aquest és un dels punts que confirma els grans avantatges d'aquests tipus de disseny, mitjançant maquinari programable.

Com es va veure a la figura 8-5, disposa de dues entrades, una per al senyal de rellotge de 102,4 KHz i una altra per a omplir el registre amb el byte de posició.

Aquest últim pot canviar sense cap problema en qualsevol moment, ja que no afectarà al posicionament. Si canvia abans d'haver finalitzat el comptatge de tot el pols, s'estarà anticipant al canvi, cosa que ja ens va bé. Si canvia més tard, simplement estarà mantenint el senyal PWM, cosa que deixa el servo immòbil en la seva posició.

Per tant, el demultiplexor va omplint els registres de posició, i aquest mòdul realitzen independentment la seva tasca. Veiem-ho en processos:

D'una banda tenim el comptador de 11 bits, això sí, aquest cop funcionant amb el nou rellotge **clk2** a 102,4 KHz:

```
process (clk2)
begin
    if clk2 'event and clk2='1' then
        if comptall=2047 then
            comptall<=0;
        else
            comptall<=comptall+1;
        end if;
    end if;
end process;
```

De l'altra, veiem un dels 8 comparadors, que ens generen un dels senyals entrants a la porta NOR:

```
process (Byte_Pos_Reg0,ent_A_comp0)
begin
    if ent_A_comp0(7 downto 0)>=Byte_Pos_Reg0 then
        major0<='1';
    else
        major0<='0';
    end if;
end process;
```

I finalment, veiem les 8 portes NOR, amb les seves entrades i amb les corresponents assignacions del senyal PWM de sortida:

```
PWM0<=not(ent_A_comp0(10) or ent_A_comp0(9) or ent_A_comp0(8) or major0);  
PWM1<=not(ent_A_comp1(10) or ent_A_comp1(9) or ent_A_comp1(8) or major1);  
PWM2<=not(ent_A_comp2(10) or ent_A_comp2(9) or ent_A_comp2(8) or major2);  
PWM3<=not(ent_A_comp3(10) or ent_A_comp3(9) or ent_A_comp3(8) or major3);  
PWM4<=not(ent_A_comp4(10) or ent_A_comp4(9) or ent_A_comp4(8) or major4);  
PWM5<=not(ent_A_comp5(10) or ent_A_comp5(9) or ent_A_comp5(8) or major5);  
PWM6<=not(ent_A_comp6(10) or ent_A_comp6(9) or ent_A_comp6(8) or major6);  
PWM7<=not(ent_A_comp7(10) or ent_A_comp7(9) or ent_A_comp7(8) or major7);
```

Amb els processos descrits en aquest capítol s'ha vist par del codi VHDL que descriu el maquinari implementat dins la FPGA. La resta, la funció del qual és definir els senyals interns i llibreries de treball es pot trobar en l'annex d'aquest document.

8.7. PROCÉS d'ESTABILITZACIÓ AUTOMÀTICA

Fins ara hem vist tot el disseny que es correspon amb el que hem anomenat procés de control de l'aeronau. No s'ha parlat encara de la segona i important part del projecte, que és la que estabilitza de forma autònoma el vehicle.

L'objectiu d'aquest segon mode de funcionament, és el de utilitzar de forma automàtica els controls implicats en els moviments transversals i longitudinals de l'aeronau, responent a la correcció d'una posició prèvia, llegida a través del sensor, l'acceleròmetre.

Per portar-ho a terme, el microcontrolador llegirà de forma rutinària la posició a través del sensor i, en cas d'estar l'estabilització automàtica activada, actuar sobre aquells controls que prèviament estava utilitzant el pilot. Per fer-ho, s'utilitzaran els mateixos mecanismes de control de programari que s'utilitzen en el mode manual, havent prèviament modificat la trama amb el nou valor calculat per a l'estabilització.

En primer lloc, mirem matemàticament el procés a seguir en la correcció.

8.7.1. Funció de correcció

Cal saber primerament, què fer amb les mesures del sensor. És a dir, quant i com s'ha de variar la posició dels servos en funció de les lectures, en funció de la inclinació en la que es troba el vehicle.

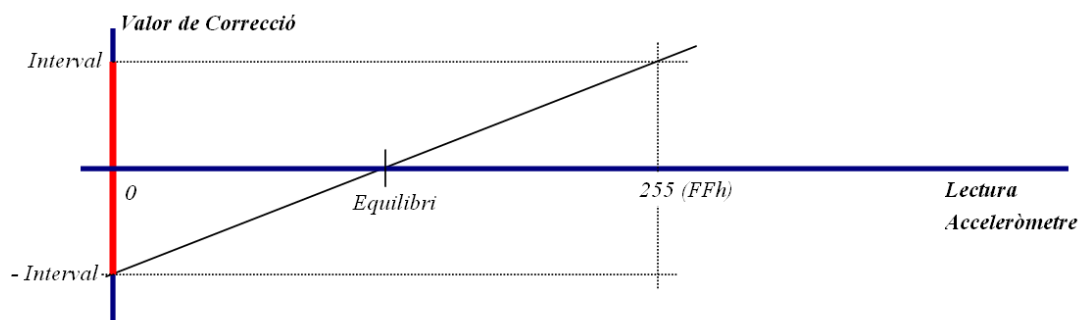


Figura 8-9. Funció genèrica per a la correcció de l'estabilització.

En la figura 8-9 es mostra com es pretén realitzar la correcció, actuant sobre la sortida del sistema, a partir d'una entrada donada. En principi, es pot considerar lineal, segons les dades del fabricant. Caldria només aplicar la funció

$$\text{Correcció} = \frac{2 \cdot \text{Interval}}{256} \cdot \text{Lectura} - \text{Interval}$$

La idea serà associar un valor de resposta a una entrada. Aquesta resposta haurà d'estar delimitada i fixada per nosaltres, en funció del valor màxim de correcció que anomenarem **interval**. Aquest valor es determina experimentalment en funció del tipus d'aeronau i de l'agressivitat o suavitat amb la que volem fer les correccions. Per al nostre cas, fixarem un valor a priori estàndard, no més gran que el màxim possible recorregut de l'actuador.

El punt d'equilibri es correspon amb la posició totalment horitzontal i estable del sensor, també donada pel fabricant. Valors cap a la dreta de la gràfica, ens determinen correcció en un sentit i valors cap a l'esquerra, una correcció a la inversa.

Aleshores, cal trobar experimentalment els diferents valors que ens dona l'acceleròmetre, és a dir la nostra entrada, quan aquest és sotmès a les diferents posicions en les que es pot trobar.

Idealment, corresponent amb les dades de 8 bits que llegim del sensor, es voldria tenir un interval de lectura entre 0 i 255 (FFh). Però la lectura real no es correspon amb això. Experimentalment s'obté el següent resultat, donant els valors directament en hexadecimal:

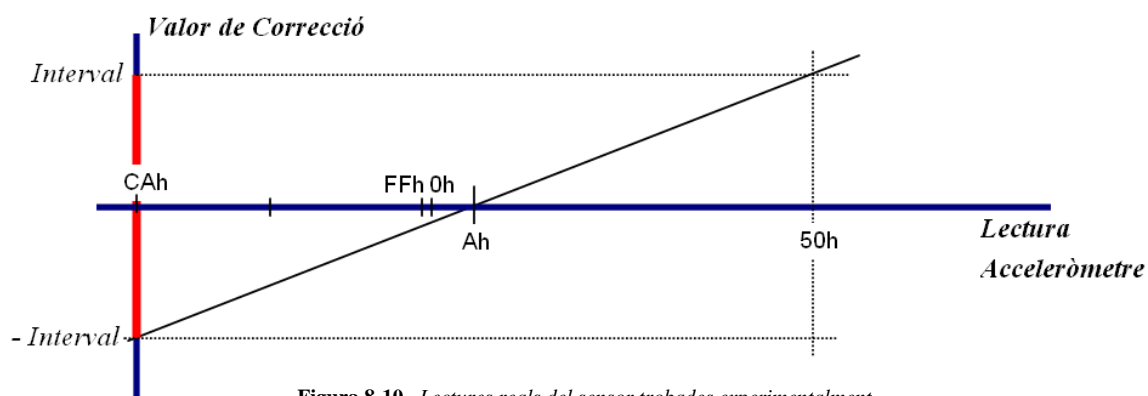


Figura 8-10. Lectures reals del sensor trobades experimentalment.

Com es mostra en la figura 8-10, aquest valor donat per l'acceleròmetre no compleix una escala consecutiva de valors que ens ajudi en la implementació matemàtica de la funció. No tenim valors “ordenats” començant per 0 i acabant en FFh com ens agradaria, en l'eix de les lectures.

Per solucionar-ho, aprofitarem el truncament que es fa automàticament en desbordar un valor de 8 bits. Per exemple

$$CAh + 36h = 100h$$

que representat en 8 bits es correspon amb el valor 00h, precisament el que ens interessa. Aplicat això a tots els valors de lectura s'obté l'escala entre 0 i FFh que necessitem.

Notem també com el total de valors que es llegeixen en totes les inclinacions possibles, és aproximadament 128 valors, en lloc dels 255 que n'havíem previst a 8-9. Diem aproximadament, perquè les lectures s'han obtingut experimentalment i la diferència $50h - CAh$ conté 134 valors.

Per tant, anomenant *novaLectura* (de 8 bits) al valor de *lectura* + 36h obtenim la següent funció a implementar:

$$Correcció = \frac{2 \cdot Interval}{128} \cdot novaLectura - Interval$$

Aquesta serà la funció que s'implementarà per a cada un dels dos eixos, en el càlcul de l'estabilització.

8.7.2. Inicialització de l'acceleròmetre

Per tal d'obtenir les lectures del sensor de posició, ens hem de comunicar amb ell mitjançant el bus I²C.

Abans però, és molt important haver configurat o inicialitzat correctament el dispositiu, tal i com s'havia vist al capítol 7.3.4. Per fer-ho utilitzem el codi que mostrem i expliquem a continuació:

```
// ***** Inicialitzo acceleròmetre *****  
  
i2c_start();  
i2c_write(0x3a);  
i2c_write(0x16);  
i2c_write(0x05);    // Escribim 0101 al $16 (Measurement Mode + 2g)  
i2c_stop();
```

Expliquem-ho comentant el protocol I²C vist a la figura 7-28, corresponent a l'escriptura d'un únic byte.

En primer lloc, el microcontrolador crida a través del seu bus al dispositiu amb el que es vol comunicar. Pensem que en un bus n'hi podrien haver més d'un. Per fer-ho recordem que l'adreça del nostre sensor és \$1D.

Després d'un **start**, s'envia l'adreça del dispositiu en 7 bits seguida d'un '0' indicant que el que voldrem fer serà escriure (per inicialitzar-lo). Així

$$0011101 \text{ \& } 0 \rightarrow 00111010 = 3Ah$$

A continuació s'envia l'adreça del registre (8 bits) sobre el que volem escriure, en el nostre cas, segons el fabricant el **\$16**.

I finalment enviem la dada que es vol escriure en el registre prèviament seleccionat, que en aquest cas és 05 donat que configurem els valors 0101 corresponents a *mode mesura* i 2g de rang de lectura.

Un cop acabada la comunicació el controlador ho indica al bus amb un **stop**.

8.7.3. Lectura de l'acceleròmetre

La lectura es fa de forma cíclica, dins el bucle infinit i principal del programari del microcontrolador, utilitzant el mateix protocol I²C.

```
while(1)
{
    // ***** LLEGIM I2C PER L'ACCELERÒMETRE *****

    i2c_start();
    i2c_write(0x3a);           // Adreça $1d + bit lectura =1 --> 0x1d
    i2c_write(0x06);

    i2c_start();
    i2c_write(0x3b);
    dadaX=i2c_read(1);
    dadaY=i2c_read(0);
    i2c_stop();
}
```

En aquest cas, s'utilitza el protocol vist a la figura 7-27, corresponent a la lectura de múltiples bytes, perquè llegirem la posició de l'eix X i de l'eix Y.

Després de l'**start**, de nou s'envia l'adreça del dispositiu, per escriure quin registre es vol llegir, el **\$06**. Posteriorment, després d'un segon **start**, s'inicia la seqüència de lectura. En aquest cas, el valor a especificar és

$$0011101 \& 1 \rightarrow 00111011 = 3Bh$$

contenint l'adreça del dispositiu seguida d'un '1' per indicar lectura.

Posteriorment es guarden en les variables *dadaX* i *dadaY* les dades enviades pel sensor.

Aquestes seran les variables que s'utilitzaran dins la funció vista al capítol 8.7.1, com a entrades per a calcular la sortida del sistema.

8.7.4. Càlcul d'estabilització i modificació de la trama

El procés de càlcul de l'estabilització es realitza aplicant la correcció matemàtica vista al capítol 8.7.1, dins també del mateix bucle principal del programa.

Quan la opció d'estabilització és activa, mitjançant el comandament del pilot, es calcula la correcció a realitzar en funció de la posició llegida. Així s'obtenen dos nous valors per a ser inserits dins la trama en els bytes corresponents als servos a actuar.

Abans però, s'aplica un filtre a les dades llegides. Aquestes dades no son estables tot i mantenir-se l'acceleròmetre en una posició fixa. Per això, es compara la dada llegida amb l'anterior i només és utilitzada si supera un límit establert per nosaltres.

També aleshores es fa l'adaptació d'escala explicada al capítol 8.7.1.

```
if (ABS(medX-dadaX)>5)
{
    medX=dadaX;
    novaX=dadaX+54;
}
```

Veiem l'exemple en un dels eixos, en concret l'eix X.

```
if (trama[7]==0x4)
{
    Estabilitza=1;
}

if (trama[7]==0x8)
{
    Estabilitza=0;
}

// ***** CALCULEM CORRECCIO D'ESTABILITZACIO *****

if (Estabilitza)
{
    CorreccioX=((int16)200*Interval/128)*(int16)novaX/100)-Interval;
    trama[2]=CentreX+CorreccioX;
```

```

    trama[3]=CentreX-CorreccioX;
}

```

En primer lloc es comprova el valor de *trama[7]*, canal assignat a l'activació o desactivació de l'estabilització.

Si està activat (*Estabilitza=1*), es calcula la correcció aplicant la funció de 8.7.1. en funció del valor immediatament llegit del sensor.

Per estalviar recursos de memòria en el microcontrolador, en lloc d'utilitzar una variable en coma flotant (float), s'ha preferit utilitzar enters. Per no perdre informació en el procés de truncament dels decimals dels valors calculats, es multiplica el valor en qüestió per un factor, **100** en el nostre cas. Posteriorment es torna a dividir per aquest factor, obtenint així el valor final.

Finalment, només quedarà substituir els valors de les trames enviats des de terra, per el valor calculat en el procés d'estabilització, omplint *trama[2]* i *trama[3]* en el cas d'estabilització lateral (alaveig) segons l'eix X.

El valor final serà

$$trama[2,3] = centreX \pm correccióX$$

Aquest valor *centreX,Y* ve determinat per nosaltres en funció de la posició central de l'actuador que és sempre la mateixa i és a més configurable. En el nostre cas és definida com a 0x7D (veure figura 8-1).

De la mateixa manera es realitza la correcció en l'eix Y, amb la diferència de que només hi haurà un servo implicat en aquesta correcció, el del timó de profunditat concretament.

$$trama[4] = centreY \pm correccióY$$

Es pot veure el codi complert en l'annex d'aquest document.

Posteriorment, mitjançant el mateix mètode utilitzat en el control manual vist a 8.4.1, aquesta trama, ara modificada, és enviada a la FPGA.

9. CONCLUSIONS I LÍNIES FUTURES DE RECERCA

L'elaboració d'aquest projecte fi de carrera ha permès treballar i aprofundir en alguns dels aspectes vistos durant la formació. Més en concret, donat el privilegi de l'elecció del tema, s'ha pogut treballar i aprofundir d'una forma molt considerable en aquells aspectes que més motivació em desperten.

En el tema en qüestió, s'ha vist la dificultat d'integrar diferents subsistemes en un de sol. Però potser aquesta és una de les funcions dels enginyers.

Les possibilitats que ens ofereixen el microcontroladors i el maquinari programable son inqüestionables. Permeten desenvolupar sistemes que haurien estat veritables reptes si s'haguessin d'implementar amb l'electrònica dels components discrets.

D'altra banda els microcontroladors estan farcits de complements que ajuden a la interconnexió entre diferents subsistemes, d'una forma molt senzilla. És curiós com el disseny electrònic s'està convertint moltes vegades en escriptura de codi.

Pel que fa al nostre sistema en concret, cal tenir molta cura dels temps dels processos. El principal topall amb el que hem donat ha estat el de no tenir control estricte sobre els temps de procés de l'ordinador. Cal per tant, millorar aquest aspecte amb sistemes de temps real.

De cara a futures línies de desenvolupament, al llarg del text han anat quedant comentades moltes d'elles. De forma immediata, un cop superats els impediments temporals, el primer que es pot fer es afegir la comunicació bidireccional.

De fet, el nostre maquinari ja està preparat perquè s'han connectat ambdues línies de transmissió TX i RX en els dos extrems. Fins i tot s'ha utilitzat aquesta circumstància durant el desenvolupament. Només és necessari aleshores implementar les noves funcions mitjançant programari.

Des del punt de vista del vehicle aeri, tasques com afegir sensors extres poden ser molt interessants per a facilitar el control.

Per exemple un GPS es connecta molt fàcilment amb un microcontrolador a través del port sèrie. Molts sensors de corrent o tensió, tenen sortides analògiques que també poden ser tractades mitjançant els convertidors AD dels microcontroladors.

Tota aquesta nova informació recollida, pot molt fàcilment ser enviada a l'estació de terra de forma que el programa de control pugui presentar o utilitzar aquesta informació. S'hauria aleshores implementat un sistema de telemetria.

Sense sortir encara del vehicle, es pot dotar de més intel·ligència al propi processador. Tenint com té les dades del GPS, a més del control en la estabilització, pot programar-se per a realitzar vols concrets de forma autònoma sense dependre de la constant comunicació amb la base.

Una altra font de recerca es troba en el programa de l'estació base. Un cop ja es tenen suficients dades i el vehicle és capaç d'autoestabilitzar-se, el propi programari de control pot comandar tipus concrets de vol.

Per exemple, pot automàticament seguir uns waypoints per establir una zona de vigilància. Pot també volar en cercles a partir d'un punt concret o fer tornar a l'estació base el vehicle en cas necessari.

És molt interessant haver aprofundit en un camp que té tantes possibilitats i que de ben segur, en un temps molt proper, serà molt més habitual en l'àmbit civil del que ho és ara.

10. BIBLIOGRAFIA

Fabricants i venedors de components

<http://www.silabs.com/products/interface/usbtouart/Pages/usb-to-uart-bridge.aspx>

<http://www.latticesemi.com/>

<http://www.microchip.com/>

<http://www.piccircuit.com/>

<http://www.st.com/internet/com/home/home.jsp>

<http://www.embeddedarm.com/products/board-detail.php?product=TS-7500>

<http://es.rs-online.com/>

<http://www.lprs.co.uk/products.html>

Components i Tutorials sobre sensors

<http://www.sparkfun.com/tutorials/167>

<http://gadgetgangster.com/find-a-project?subcat=27>

http://www.chrobotics.com/index.php?main_page=product_info&products_id=2

Informació General i Tutorials

http://en.wikipedia.org/wiki/Unmanned_aerial_vehicle

<http://www.aquihayapuntes.com/indice-programacion-pic-en-c.html>

<http://www.diydrones.com/>

<http://sites.google.com/site/mikuadricoptero/home>

<http://www.aerovision-uav.com/>

<http://www.schiebel.net/>

Bibliografia

“Compilador C CCS i simulador PROTEUS para Microcontroladores PIC”

Eduardo García Breijo

Editorial Marcombo 2008

“Microcontroladores PIC”

E. Martín Cuenca, J.M. Angulo Usategui, I. Angulo Martínez

Editorial Paraninfo 2001

“VHDL Lenguaje para síntesis y modelado de circuitos”

Fernando Pardo, Jose A. Boluda

Editorial Ra-Ma 2003

11. ANNEXOS

Relació de la informació que es pot trobar en el CD annex:

- Codi VHDL de la FPGA
- Codi C del microcontrolador 16F690
- Fulls d'especificacions de components
 - Acceleròmetre MMA7455
 - Microcontrolador PIC 16F690
 - Placa Avaluació MACHXO
 - FPGA MachXO LCMX0256C
 - Transceptor de ràdio ER400TS
 - Reguladors de tensió TI uA78M00 series
- Arxius disseny de la PCB